Haaga-Helia
ammattikorkeakoulu Oy

Future Trends and Development Methods in Software Quality Assurance

Päivi Williams

Master's Thesis
Degree Programme in
Information Systems
Management
2017

Abstract

Haaga-Helia
ammattikorkeakoulu Oy

Author
Päivi Williams

Degree programme
Information Systems Management, Master's Degree

Thesis title
Future Trends and Development Methods in Software Quality Assurance

Number of thesis pages and appendix pages
59 + 5

The main purpose of this thesis is to study the future trends and development methods in software quality assurance from a testing point of view. The main scope of this thesis is to study what views and expectations customers and personnel of Comiq Ltd have about the future trends and development methods in software quality assurance, how those trends will impact the development methods in the future and what should be the focus areas of Comiq in the future to update and maintain employee's high competence levels.

The background studies of the thesis focus on current predictions in relation to agile and DevOps principles as well as test automation. Agile and DevOps are the main trends that are commonly used in software development and quality assurance is an essential part of both agile and DevOps.

The research method was a quantitative data analysis. The sample data was collected in June 2017 by conducting an online questionnaire that was addressed to Comiq customers and personnel. A total of 93 people participated. The questionnaire had a total of 15 questions relating to the properties of quality assurance experts, test automation and methods of importance in achieving future quality goals.

Based on the study results, the most important properties of quality assurance experts are understanding the customer's needs, communications skills, and continuous development of skills and understanding the business area. Test automation is currently seen as one of the most important factors in improving the quality of software. Overall quality can be assured by automating all necessary functionalities including regression testing. Test environments should be automated and version control and functional requirement management should continuously support testing.

The biggest obstacles seen by the participants in achieving the overall target state of quality assurance are lack of test automation, money, time, recourses, and current software development methods and efficient ways of working.

Test automation is an area that will continue to grow and it should be invested in more in the future. Results show that customers and personnel are aware of current trends and want to improve overall quality assurance by growing maturity of agile and DevOps and by involving testing as early as possible.

Keywords
Software quality assurance (SQA), Agile, DevOps, Test automation

Table of contents

1 Introduction

The change and growth of digitalization can be seen everywhere. New digital services are available in all types of business areas. End-users of these digital services are more and more selective what services they want to use.

As digital transformations change the world, more and more of an organization's services are exposed to direct customer interaction. This is one of the biggest factors driving testing and quality assurance towards business goals such as decreasing time-to-market, increasing security, performance and customer satisfaction. (Waayenburg 2016, 5.)

Quality assurance and testing is rapidly developing to deliver towards such goals. We see trends such as the increased adoption of agile and DevOps, the use of predictive analytics, increasing automation of quality assurance and testing practices, as well as a change in the skills required from quality experts. (Waayenburg 2016, 5.)

In the IT world, new technologies are changing and successful organizations need to follow current trends as well as future opportunities. The purpose of this thesis is to study current testing trends and conduct research in relation to Comiq customers and personnel about future expectations. The goal of this study is to find answers how current trends and development methods will impact the field of testing and quality assurance and what are the key areas to focus on in the future. The study was executed by a questionnaire that was directed at Comiq customers and personnel. The scope of the study is described in more detail in chapter 2.

The background studies focus on current software development methods such as agile and DevOps, test automation, current predictions, frameworks as well as business benefits and challenges. Background studies give an overall picture of relevant matters from a testing point of view. Background studies is presented in chapter 3.

Chapter 4 Design describes how the study was designed and how the questionnaire questions were selected and accepted. Chapter 5, Implementation describes the implementation details such as what tools were used and why a tool was chosen and when the questionnaire was conducted.

Chapter 6 Analysis describes all the results in more detail. The results of the questionnaire have been analysed individually, graphical images have been used to

represent some questions and the results of the open questions are also presented in this chapter. The analysis chapter gives specific details of each question separately and the differences between the participant groups that are represented.

Chapter 7 Discussion is author's own interpretation of the research results and answers to the research questions. Interpretation is done based on the actual results presented in the previous chapter analysis. This chapter combines the analysis of the results and the personal experience and interpretation of the author. Chapter 7 also includes the author's recommendations. Chapter 8 presents the conclusions of the whole study.

2 Scope of study

The purpose of this study is to look into more detail at the field of testing, quality assurance and the current trends and software development methods such as agile and DevOps from a testing point of view.

The theory part of this study will focus mainly on agile and DevOps and the main principles including assessment of best practices like different framework approaches. The theory part will also describe quality assurance and testing approaches like shift right testing and shift left testing principles.

Agile as a software development method has been in use a long time and the current prediction is that agile methods are moving towards DevOps. Based on this prediction, both agile and DevOps are selected to be a part of the theory background of this study. The study includes analysis of future trends and development methods related to testing and quality assurance. All software development methods have their own challenges and the theory part also describes current views of possible problem areas.

The main scope of the thesis is to study what views and expectations customers and personnel of Comiq have about future trends in the field of quality assurance. Scope is to study how current trends and development methods will impact the fields of testing and quality assurance in the future. The study also aims to collect important information about areas that Comiq should focus more (e.g. in training, studying point of view) in the future to update and maintain employee's high competence levels following these future trends, requirements and expectations that Comiq´s customers have.

Other development methods like waterfall, Lean Development, SAFe (Scaled Agile Framework) are not included in the scope. Testing tools and the use of tools are not included in the study. The study boundaries have been agreed in collaboration with Comiq.

2.1 Case Company

Comiq is a preferred DevOps, software testing and quality assurance partner for the leading companies in various industry sectors and markets. Comiq works together with the biggest banks, retailers, insurance, telecommunication and heavy industry companies. Comiq is headquartered in Helsinki and currently employs around 70 software quality assurance and DevOps experts.

Comiq specializes in test management, test automation, technical testing, DevOps and requirements specifications in large demanding business critical software development projects. For many years, Comiq has been listed among the fastest growing Finnish companies in the Deloitte Technology Fast programs and performed superbly in Great Place to Work contest.

2.2 Research questions

1. What views and expectations customers of Comiq have about future trends in the field of quality assurance?

2. How current trends and development methods will impact in the fields of testing and quality assurance in the future?

3. What should be the focus areas of Comiq (from a training, studying point of view) in the future to update and maintain employee's high competence levels?

2.3 Research method

This study will use quantitative data analysis as a research method. Quantitative research is a distinctive research strategy that is described as entailing the collection of numerical data and as exhibiting a view of the relationship between theory and research. Method uses deductive, a predilection natural science approach and objectivist conception of social reality. (Bryman & Bell 2015, 161.) Quantitative data is based on meanings derived from numbers, collection results in numerical and standardized data and analysis conducted through the using of diagrams and statistics. (Saunders, Lewis & Thornhill 2009, 204).

The purpose of the study is to gain detailed information about the views of Comiq's customers and employees and therefore quantitative data analysis is the correct research method in this study. Part of the study is to execute a customer questionnaire. The questionnaire will focus on customer expectations and views about quality assurance as well as their expectations about testing experts.

After collecting relevant theory material and executing the questionnaire the study aims to compare current trends and development methods as to how customers see the future of quality assurance.

3 Background studies

This study focuses on various aspects of software quality assurance and background studies chapter will describe official definition of software quality assurance.

Predicting future trends in technology areas that are rapidly evolving is quite often very challenging. Obviously, nothing is sure when it comes to predicting the future. However, some of most respected experts in the area of quality assurance, like Hans van Waayenburg and Raffi Margaliot, authors of world quality report 2016 – 17, have a few quite good ideas about what the future will bring. In this chapter, the main goal is to present the current trends, views, predictions that one of the biggest technology companies have about predicting about future of quality assurance.

Agile and DevOps were chosen as two main subject areas for the background study as these methods offer significant opportunity to drive towards faster release cycles compared to traditional software development models. It is evident that the issues around managing and implementing quality and test measures are an increasing challenge. Many organizations are trying to find ways to achieve speed with the suitable level of quality. (Waayenburg 2016, 9 - 10). This chapter also describes objectives, business values and challenges of agile and DevOps.

3.1 Definition of Software quality assurance

According to the International Software Testing Qualifications Board (ISTQB) the official description of quality assurance is “Part of quality management focused on providing confidence that quality requirements will be fulfilled”. (ISTQB 2017).

Described in more detail quality assurance is a set of different activities for ensuring quality in software engineering processes that ultimately result in quality in software products. Quality assurance includes activities like process definition and implementation, auditing and training. Quality assurance can be used by various processes like software development methodology, project management, configuration management, requirements management, estimation, software design and testing. The main responsibilities of quality assurance are identifying weaknesses in the processes and correcting those weaknesses to continually improve the process. The quality management system is normally based on one or more of the following most popular standards like CMMI, Six Sigma or ISO 9000. (Software testing fundamentals 2017).

Software quality assurance encompasses the entire software development life cycle and the goal is to ensure that the development or maintenance processes are continuously improved to produce products that meet specifications or requirements. (Software testing fundamentals 2017).

3.2 Current predictions of quality assurance

Digital transformation is changing the world and companies in all sectors continue to use new digital services to connect with their customers. Companies that have an ability to bring new products and services to market ahead of the competition are most likely to stand out against other competitors. Therefore, agile methodology is firmly embedded in development lifecycles and this trend looks set to continually grow in the future. (Buenen & Muthukrishnan 2016, 5).

According to Capgemini World Quality Report, quality assurance aims more and more to improve business outcomes such as business experience, revenue growth and uninterrupted business operations. Quality assurance is no longer just getting the product to market as quickly as possible and finding as few defects as possible. The new trend is more about transparent business risk profiling and the shifting left of quality responsibilities as well as upgrading critical quality assurance components such as test data, test environment and test network. This new trend will drive effective decision making and deliver business results. (Buenen & Muthukrishnan 2016, 6).

Agile and DevOps offers significant opportunity to drive faster release cycles and it is evident that issues around managing and implementing the quality and test measures are an increasing challenge also the issues are occurring more often. It is challenging for organizations to find out the answers to questions like how to achieve speed with the right level of quality. This means it is important to remove boundaries between disciplines and integrate quality validation and testing activities in to the agile development lifecycle. The current prediction is that agile as well as DevOps will continue to grow. The adoption of quality assurance is making a corresponding movement. Organizations are challenged to achieve the right speed with the right level of quality. (Buenen & Muthukrishnan 2016, 7-9).

One question to consider; is the agile method the right approach when the main goal is improving business outcomes? Agile as a software development method has been around us for years already. Edmondson (Edmondson 2015) describes that the end goal of agile

is not stand ups, burn-down charts, refinement meetings or sprint velocity. The most important end goal has always been to deliver value. Agile is simply a tool to help us accomplish the job. If organizations keep focusing on the tool they will more easily miss the point and the value is not delivered and job is left undone. (Edmondson 2015.)

Based on World Quality Report 2016 – 17, test environment management and test data management need to evolve as DevOps drives further virtualization and digital transformation makes itself felt. There is an increasing trend for enhancing and increasing the flexibility of test environments across all areas of testing. There is still a lack of skillsets in test environment management at a broader information technology industry level although there are signs of increasing maturity in these skill areas. Organizations that invest in enhancing their test environments, are better equipped and able to adopt agile and DevOps practices at the level needed to secure business outcomes and drive more customer value. (Walgude 2016, 45).

The amount of the total information technology budget that is allocated towards testing and quality assurance will largely control how well overall quality assurance can be verified. Predicted and realised testing and quality assurance budgets are presented in the figure below. The steady yearly growth of quality assurance and testing budget has come to a halt. On average the organizations are now spending 31% of its information technology budget on testing and quality assurance and the prediction for the next two years is higher. Figure 1 demonstrates quality assurance and testing budget allocating as percentage of IT spend. (Shea 2016, 51).

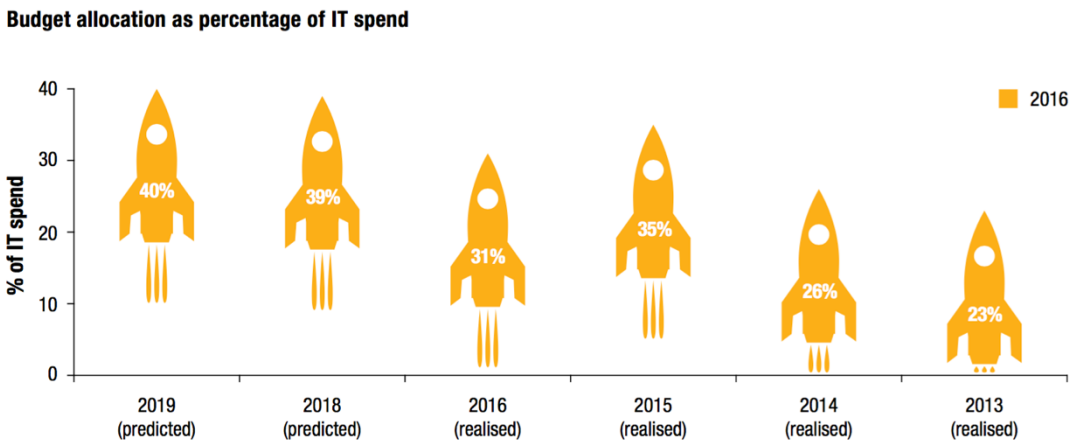


Figure 1. Quality assurance and testing budget allocation as percentage of IT spend (Shea 2016, 51)

3.3 Test Automation as part of quality assurance

Organizations are moving toward digitalization of their businesses at same time as development teams are looking for ways to optimize the delivery of applications to the point of a continuous flow of releases into production. Automation is an indispensable element in these efforts and software test automation tools are more and more the part of a larger tool chain that enables DevOps. The test matrix grows exponentially while the demand of faster delivery of software rises. Same time delivery methods and client endpoints get ever more varied. (Herschmann & Murphy 2016).

Quality assurance in organizations simply cannot continue to maintain the speed due to a heavy reliance on manual testing processes, deficiency of skills, insufficient resources, and an inability to scale the technology and processes. These challenges are what software test automation is designed to solve. The key is to know what the state of the build is and to ensure via an automated regression suite that the latest update has not broken systems. This will reduce overall costs and can boost confidence. (Herschmann & Murphy 2016).

Organizations are still struggling to move from the tactical task of testing software as part of a distinct phase or activity (often done manually) to a more automated form of quality assurance that essentially requires no human intervention. The cost of both - creating and maintaining automation - is often high compared with the cost of performing the test manually. More value to the automation is gained through the ability to run the same tests on several different platforms, devices or configurations. Automation is all about being able to create and maintain tests that are specially designed to cover business-critical risks and satisfy real business requirements. Maintenance test should be done with as little effort as possible and test should be able to run as often as needed in a stable test environment. (Herschmann & Murphy 2016).

3.4 Principles of agile

Quality assurance is used to ensure software product quality as a part of the development process. It is a crucial component of most software development efforts. Testing is one part of quality assurance and is the process of analyzing a software item to detect the differences between existing and required conditions. Testing is also evaluating the features of the items. (Itkonen, Rautiainen & Lassenius 2005).

Traditionally testing has included a set of activities at the end of the development project, however this has often unpredictable length and effort requirements. Therefore time-boxed agile projects require a different approach to quality assurance. (Itkonen al. 2005).

Agile is a common term for methods and practices that have emerged over the past two decades to increase the relevance, flexibility and business value of software solutions. These approaches are meant to solve the problems that have historically emerged in different software development activities in the IT industry such as budget overruns, missed deadlines, low quality outputs and dissatisfied users. Agile methods are practical approaches for applying the fixed resources of an organization to deliver high business-value software solutions. (Cooke 2016, 16).

Agile software development methods are based on iterative and incremental development. Development method is using short time-boxed development cycles. The priority agile methods are to satisfy the customer through early and continuous delivery of valuable software. The idea behind agile software development is to highlight individuals and interactions, customer collaboration and responding to change. (Itkonen al. 2005).

The most common agile methods include iterative strategies for managing software development projects. These are e.g. Scrum, Dynamic Systems Development Method (DSDM), Feature-Driven Development (FDD), the Rational Unified Process® (RUP), and the Agile Unified Process (AUP). Methods include strategies for optimizing software development work, such as Extreme Programming (XP) and Lean Development and strategies for managing software development and maintenance projects such as Kanban and Scrumban. Extensions of agile methods are supposed to support large enterprise-wide teams and shared corporate objectives. Example of these strategies are Scaled Agile Framework (SAFe), Scrum of Scrums, Large-Scale Scrum Framework (LeSS) and Nexus. (Cooke 2016, 17).

These agile methods have been successfully used by several organizations all over the world. To fully appreciate the effectiveness of agile methods, it is worthwhile taking the time to understand the business environment that caused these approaches to be established in the first place. (Cooke 2016, 18).

3.5 Agile Manifesto

Agile manifesto was created in 2001. A group of 17 people representing the most widely used lightweight software development methodologies agreed on a common set of values

and principles which became known as the Manifesto for Agile Software Development or the Agile Manifesto. Four main statements of Agile Manifesto are listed below. (Black al. 2014, 9).

> **Individuals and interactions** over processes and tools
> **Working software** over comprehensive documentation
> **Customer collaboration** over contract negotiation
> **Responding to change** over following a plan
> (Black co 2014, 9).

Agile Manifesto main 12 principles are

1) The highest priority is customer satisfaction through early and continuous delivery of valuable software.

2) Changing requirements are welcome at any time of the development. Agile processes harness change for the customer's competitive advantage.

3) Working software is delivered frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4) Business people and developers work together daily throughout the project.

5) Project is built with motivated individuals. They are given an environment and support they need and they are trusted.

6) Communication in development team is mostly done face-to-face because that is the most efficient and effective method of conveying information.

7) Primary measure of the progress is working software.

8) Agile processes promote sustainable development. The sponsors, developers, and users can maintain a constant pace indefinitely.

9) Continuous attention is paid to technical excellence and good design enhances agility.

10) Simplicity is essential. The art of maximizing the amount of work not done.

11) The best architectures, requirements and designs emerge from self-organizing teams.

12) At regular intervals, the team reflects on how to become more effective and then tunes and adjusts its behaviour accordingly. (Agile manifesto 2001).

3.6 Principles of DevOps

The word "DevOps" is a combination of the words "development" and "operations". DevOps is a practice where collaboration between different disciplines of software development is encouraged. DevOps roots come from agile software development principles and DevOps can be said to relate to the first principle “Individuals and interactions over processes and tools”. (Verona 2016).

DevOps is not just another software development methodology. DevOps is more a way of thinking and a way of working. DevOps is a framework for sharing stories and developing empathy, enabling people and teams to practice their crafts in effective and lasting ways. It is part of the cultural weave that shapes how we work and why. (Davis & Daniels 2016).

The core idea behind DevOps is to remove the boundaries between the traditional silos of business, IT development, quality assurance and IT operations, and to extend the agile principles. Buenen and Muthukrishnan believe that this process can only be successful if the quality assurance aspect is integrated and maximum automation is achieved in each step of the DevOps lifecycle. (Buenen & Muthukrishnan 2016, 12).

Collaboration between teams from a practical point of view, could be use of a common system to report bugs. Quite often development teams, quality assurance teams and operations teams use different systems to handle tasks and bugs. This creates unnecessary conflict between the teams and further separates them when they should really focus on working together. (Verona 2016).

One of the main goals of DevOps is automation and continuous delivery. Automating repetitive and tedious tasks leaves more time for human interaction, where true value can be created. (Verona 2016.)

DevOps is all about technical practices like continuous integration and continuous development, test automation and infrastructure automation and the main difference from agile is that agile concentrates more about actual processes. Figure 2 shows DevOps process versus agile process. (Vashishtha 2017).

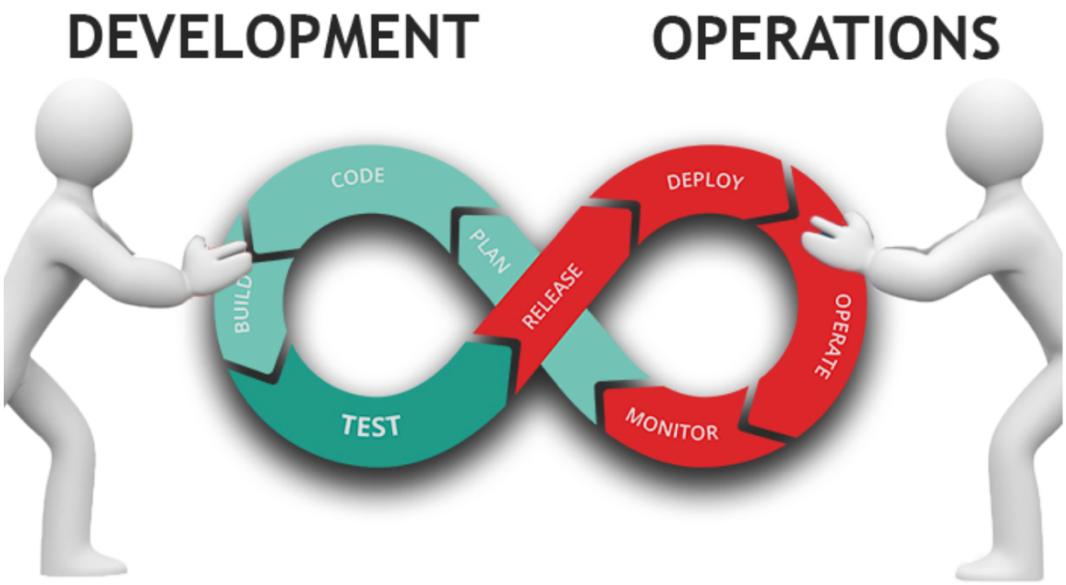


Figure 2. DevOps versus agile process (Vashishtha 2017)

DevOps is also about following agile principles and in addition including systems and operations instead of stopping its concerns at code check-in. Apart from working together as a cross-functional team of designer, tester and developer as part of an agile team, DevOps recommends adding operations as well in the definition of a cross-functional team. DevOps emphasises breaking down barriers between developers and operations teams and getting them to collaborate in a more beneficial way. (Vashishtha 2017).

3.7 DevOps maturity model

DevOps maturity model describes organization's DevOps maturity in different levels. Initial level means that organization has poor and ad-hoc communication and coordination, no automation and processes are unpredictable, uncontrolled and reactive. The managed level means that collaboration is more managed, automation is solid and processes are managed but not standardized. The defined level means that collaboration includes shared decision making and is accountable, centralized processes are automated across the application lifecycle and processes are standardized across the whole organization. The measured level means that collaboration processes are measured to identify inefficiencies and bottlenecks, automation processes are analysed and measured against the business goals. The optimized level is the highest level and the level that all organizations should aim to achieve. Collaboration at the optimized level is effective knowledge sharing and individual empowerment. Automation is self-serviced, self-learning, uses analytics and self-remediation. Processes risks and costs are optimized.

The maturity model is one tool that organizations can use for measuring the DevOps maturity. Figure 3 shows the different levels of DevOps maturity. (Topham 2013.)

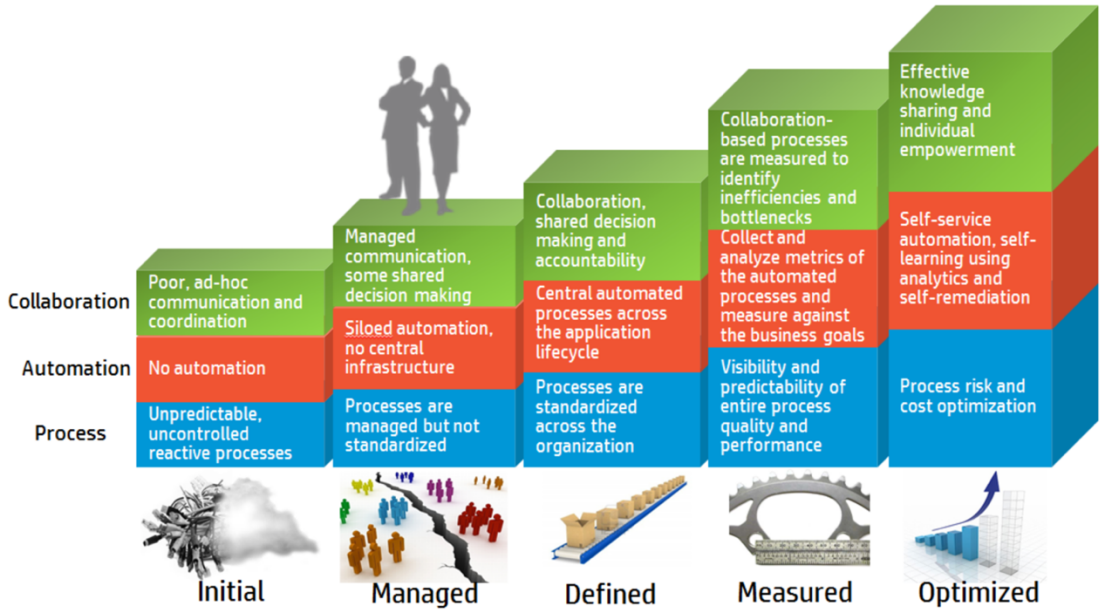


Figure 3. DevOps Maturity Model (Topham 2013)

3.8 Shift left testing

The definition of shift left testing refers to a practice in software development where teams focus more on quality, work on prevention instead of detection, and begin testing earlier than ever before. The main goal is to increase quality, shorten long test cycles and reduce the possibility of unpleasant surprises at the end of the development cycle. In many organizations, automated testing of today's integrated applications is being executed via the user interface, after the complete application has been developed and deployed. (Wagner 2015).

If organizations wait until all the pieces become available before testing, commences often causes delays, adds risk to the project or results in discovery of late stage defects. The overall idea behind shift left testing is to help to avoid rework and delays that can occur when major defects are discovered late in the testing cycle. Shift left testing aims to avoid issues by performing integration tests as soon as possible. (Wagner 2015).

One practical alternative option is to build virtual services to mock the associated components in case all the components are not available to test at the time. Agile development teams in the recent years have been able to minimize the risk associated with defect isolation by delivering working code in small batches and development teams

might not always start developing the most technically challenging parts of the code which includes complex integrations to associated components. According to shift left testing ideology teams that continuously test their code against integrated components can have better outcomes in reducing risk, as they are better able to find and fix potentially disruptive defects earlier in the lifecycle. (Wagner 2015).

Continuously delivering, building and testing in a tightly operationalized fashion allows the delivery team to receive feedback on code quality continuously. The benefit of this is that delivery teams can speed quality products to the market and validate new products or features with the customer. (Wagner 2015). Figure 4 shows the overall quality and Shift-Left testing life cycle.

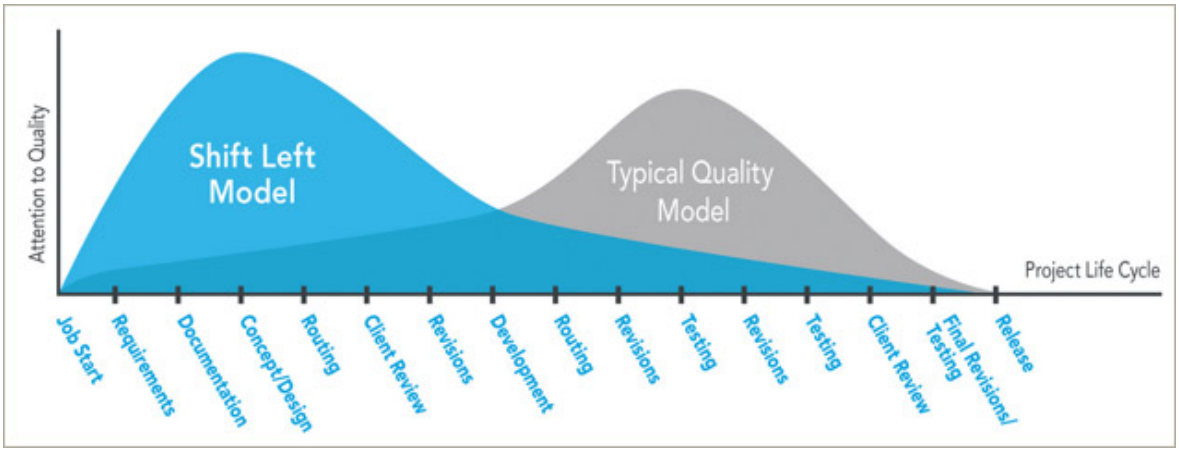


Figure 4. The overall quality and Shift-Left testing life cycle (Hanspal 2014)

3.9 Shift right testing

Wagner's definition of shift right testing is simply testing software in production. Shift right testing ideology is making a conscious decision to carry a certain level of risk deploying to production so that the business can get end user feedback faster and beat their competition to market. Delivery teams can also validate a hypothesis they are experimenting with. (Wagner 2016).

The level of risk in shifting right can be lowered by adopting practices such as dark launches, implementing business toggles, or executing canary deployments. These terms all refer to limiting the user group who have access to the latest version of the application. (Wagner 2016).

Dark launches mean that a new software version or feature is visible only for specific testers or humans interacting with the system. After tester's approved test results the new

version is made visible to everyone. Dark launch strategy should only be considered for performance testing mainly because strategy allows performance testing in the real production environment. (Münch & Mäkinen 2013, 49).

The idea behind building a business toggle is to have toggles to turn some feature on or off. This is relatively straight forward and can be implemented in the code level. It does not need any special tools or processes to be achieved. (Münch & Mäkinen 2013, 49).

Canary deployment is a strategy where the release of a software is released first only to a small group of users. This can be done automatically by the help of release management systems. After new version has been determined that there are no problems the release management system can automatically (or via a manual step) release the latest build to all the users. (Münch & Mäkinen 2013, 50).

One of the key recommendation of Buenen and Muthukrishnan is to begin the agile and DevOps journey by first shifting test left to involve the test teams right from the start. They recommend starting shift testing right after achieving scale and building continuous testing, environment virtualization capabilities, cloud-based environments and predictive analytics capabilities for DevOps. (Buenen & Muthukrishnan 2016, 12).

3.10 Different framework approaches

The main principle of quality assurance is that testing is integrated throughout the entire lifecycle. This means that all types of quality assurance tasks need to be implemented early and continuously because critical defects found at the end can seriously derail delivery or quality of the product. Different frameworks use a slightly different approach to quality assurance and testing and these approaches are described in more detail below. (Berridge al. 2015, 65.). All approaches below (except Specification by example) rely on automated testing and effective continuous integration (Berridge al. 2015, 67.)

3.10.1 Test First Development (TFD)

TFD practice means that tests are written before any development or coding to meet story acceptance criteria. When a story is to be developed, all team members get together and develop the tests prior to delivery. It is practical if the customer is involved to create tests so they can verify that the acceptance criteria are met. After tests have been written, build cycle will continue until all defects are fixed and the customer has accepted the story. This practice validates that what has been built meets the story acceptance criteria. (Berridge al. 2015, 65 – 66.)

3.10.2 Test Driven Development (TDD)

TDD is implemented at the unit or component testing level. This validates that what has been built passes the tests and meets the acceptance criteria upon which the tests are based. TDD also validates that the design is appropriate with minimal technical debt. The idea is that the team focuses on conditions in the test that could possibly cause the code to fail. The development is said to be complete only after there are no more failure conditions. TDD has many benefits, including good design principles, speed of test-code cycle, and focuses on the interface of the code. The unit test documents the expected behavior of the code, is repeatable and can be automated. (Berridge al. 2015, 66 – 67.)

3.10.3 Acceptance Test Driven Development (ATDD)

Acceptance Test Driven Development (ATDD) is like TDD but closer to user acceptance testing (UAT) (Berridge al. 2015, 67.) Based on ISTQB, ATDD is a test-first approach in which acceptance criteria is well understood by the development team and test cases are created based on the bullet points stated in the acceptance criteria. Modification to the acceptance criteria may be done prior to the planning of negotiation meeting with the product owner. (ISTQB 2017).

ATDD involves team members with different perspectives; like customers, developers and testers to collaborate to write acceptance tests in advance of implementing the corresponding functionality. Acceptance tests represent the user's point of view and act as a form of requirements to describe how the system will function. ATDD serves as a way of verifying that the system functions are as intended. (Agile Alliance 2017).

3.10.4 Behavior Driven Development (BDD)

In BDD the focus is on scenario testing and the goal is to verify that the system behaves in the way the user expects it to behave. BDD provide a very effective bridge between people with analytical, design, coding and testing skills because it encourages people to work together as a successful team instead of passing documents to each other. BDD goes beyond simply delivering software that is working and designed well. (Berridge al. 2015, 67.)

3.10.5 Specification by example

Specification by example verifies that the customer’s requirements match what is being implemented. Testing focuses on the parts of the system that create the greatest business

value. Specification is defined by Gojko Adzic as a set of process patterns that facilitate change in software products to ensure that the right product is delivered efficiently. By product Adzic means software that delivers the required business effect or fulfills a business goal set by the customers or business users and is flexible enough to be able to receive future improvements with a relatively flat cost of change. (Berridge al. 2015, 67 – 68.)

The key elements to specification are e.g.

- Derive system scope from business goals that are clearly expressed and understood by both, the customer and stakeholders
- Specify acceptance criteria for stories collaboratively between team, customer and stakeholder, as well as agreeing appropriate levels of testing
- Illustrate and agree requirements by using examples
- Refine specifications throughout the lifetime of the product
- Automate validation without changing specifications
- Validate frequently that the product being built meets the specifications
- Evolve a documentation system that consists of simple to understand requirements and tests that prove them.
  (Berridge al. 2015, 67 – 68.)

3.11 Objectives of different development methods

All software development and project delivery approaches share the same basic objectives:

1. Replace large up-front investment in solutions where incremental investment is based on business value returns

2. Give more attention to project team members on delivering capabilities. This will help generate the highest business value for the organization

3. Encourage communication between business and IT development to increase the relevance, usability and quality of delivered software

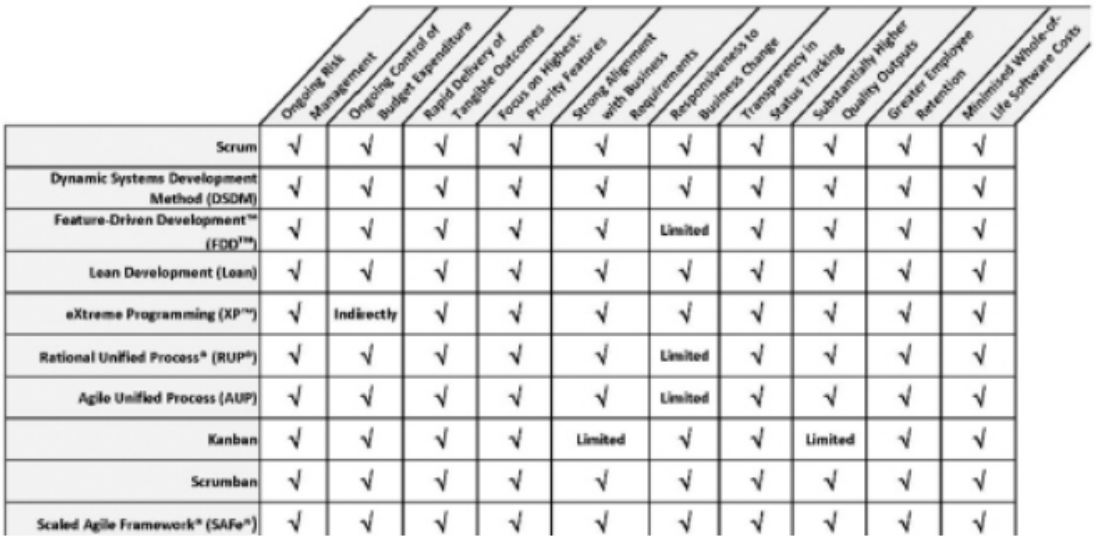
4. Trust the team members to deliver value

5. Verify that software solutions are responsive to ongoing industry, organizational and technology changes. (Cooke 2016, 16-17).

3.12 Business value of agile and DevOps

Agile as a software development method can create high business value to almost any organization. It is obvious that organizations should adopt any used method before use of method can bring more business value to the organization.

The business value that agile methods can provide to organizations can be refined into ten core agile business benefits as mentioned in figure 5.



	Ongoing Risk Management	Ongoing Control of Budget Expenditure	Rapid Delivery of Tangible Outcomes	Focus on Highest-Priority Features	Strong Alignment with Business Requirements	Responsiveness to Business Change	Transparency in Status Tracking	Substantially Higher Quality Outputs	Greater Employee Retention	Minimised Whole-of-Life Software Costs
Scrum	√	√	√	√	√	√	√	√	√	√
Dynamic Systems Development Method (DSDM)	√	√	√	√	√	√	√	√	√	√
Feature-Driven Development™ (FDD™)	√	√	√	√	√	Limited	√	√	√	√
Lean Development (Lean)	√	√	√	√	√	√	√	√	√	√
eXtreme Programming (XP™)	√	Indirectly	√	√	√	√	√	√	√	√
Rational Unified Process® (RUP®)	√	√	√	√	√	Limited	√	√	√	√
Agile Unified Process (AUP)	√	√	√	√	√	Limited	√	√	√	√
Kanban	√	√	√	√	Limited	√	√	Limited	√	√
Scrumban	√	√	√	√	√	√	√	√	√	√
Scaled Agile Framework® (SAFe®)	√	√	√	√	√	√	√	√	√	√

Figure 5. Ten core agile business benefits by agile method (Cooke 2016, 27)

These combined benefits can bring real productivity gains to organizations from their software development activities. They allow organizations to bring IT development work back under business control. (Cooke 2016, 25-26).

The sooner the product is delivered, the earlier the product starts to produce business value and the swifter business and technical feedback can be sought. The faster teams can inspect a product via feedback, the quicker they can adapt to deliver the appropriate stories as they evolve. Agile aims to deliver value via useable features from small delivery batch sizes (sprints) and continuous delivery makes this possible. Agile deliveries can be made within releases or projects although the longer the team waits to deliver valuable features the bigger the delivery batch size becomes and with this comes the inherent problems that causes risk of confusion, errors or low productivity. (Berridge al. 2015, 103.)

Based on Eficode, the real value of DevOps is that DevOps unifies development and maintenance and extends visibility all the way to the customer.

1) Requirements can be found in one clearly defined place
2) Development environments are known and can be provisioned automatically
3) Efficient version control enables controlled development and maintenance
4) New features are tested automatically during different stages of the process
5) Automatic quality assurance is connected to the original requirements
6) New product versions can be released either completely automatically or by the push of a button
7) Server environments are virtual or in the cloud. They can be configured and deployed automatically
8) The operation and efficiency of services can be monitored across the entire organization
   (Eficode 2017).

3.13 Challenges of agile and DevOps development methods

Agile manifesto describes the main principles for agile software development. These principles are described in more detail in chapter 3.5. From the traditional testing view point these principles include several challenges (Itkonen al. 2005).

Early and continuous software delivery with a rapid release cycle is a challenge for testing because the rapid release cycle puts fixed deadlines for testing activities. This does not allow extending the testing period if more defects are found than estimated. (Itkonen al. 2005).

Agile method also demands that we should welcome changing requirements even in late stages of development. Traditionally testing and quality assurance methods are based on specifications that are completed in a certain phase of development. After specification has been completed it can be used as a basis for test design and other quality assurance activities. When requirements can change during late phases it obviously challenges quality assurance. (Itkonen al. 2005).

Agile development emphasizes on face-to-face communication and business people and developers must work well together. Therefore, the documentation that traditional testing is based on does not necessarily exist. The newest and detailed information about the

expected results can often be in the heads of the developers and the business people and due to this it challenges quality assurance and final validations. (Itkonen al. 2005).

Because working software is the primary measure of progress, testing cannot be left as a last phase in the current iteration. Quality assurance must provide information on achieved quality early enough to enable evaluating if the produced code is working software or not. (Itkonen al. 2005).

Simplicity brings along one more challenge. Simplicity makes it challenge to keep necessary quality assurance practices included in the development process. Different quality assurance activities are easily seen as unnecessary and unproductive because they do not directly add value in terms of code and features. (Itkonen al. 2005).

Research conducted by Vanson Bourne 2013 top eight challenges of DevOps are presented in the figure below. DevOps is not seen as an easy software development method, with 35 percent reporting the key obstacle to DevOps adoption is organizational complexity. The second biggest obstacle with 28 percent refers to the lack of alignment of roles and responsibilities across development and operations. The third biggest obstacle is security matters. (Kolose 2015). Figure 6 shows the top obstacles to implementing DevOps according to Kolose.

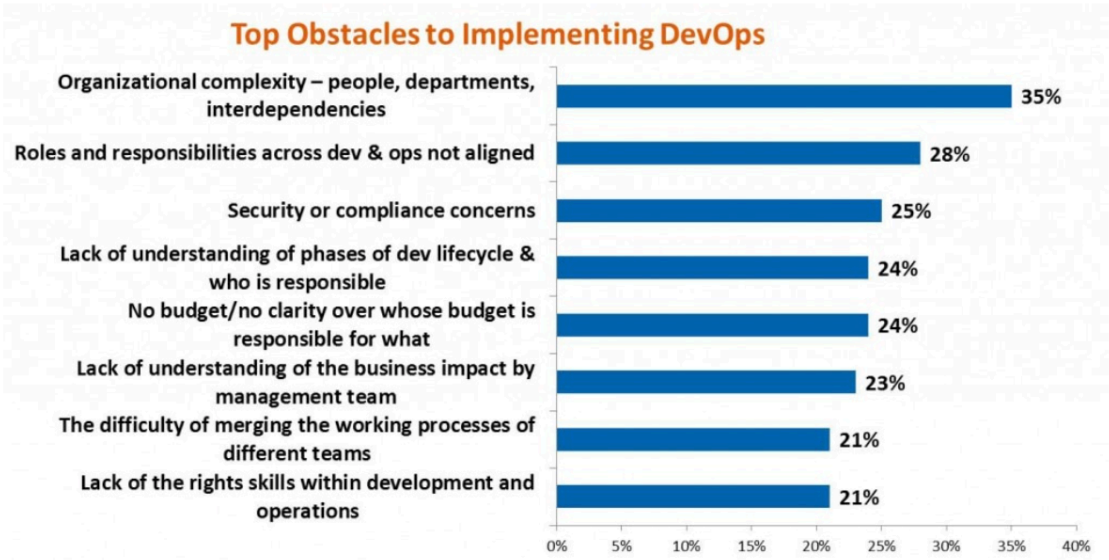


Figure 6. Top obstacles to implementing DevOps (Kolose 2015)

4 Design

The overall idea behind the study is to find out Comiq's customers and quality assurance experts views about the future trends and development methods in the field of software quality assurance. Comiq has many quality assurance experts working for different customers in several different business areas. All customers and individuals have their own strategies and methods related to quality assurance and study aims to find out how different people in different positions see the future.

This study used quantitative data analysis as a research method. Study was executed by collecting relevant theory about the current trends and development methods related to testing and quality assurance and executing online questionnaire addressed to Comiq's customers and personnel.

Draft version of the questions were planned based on current predictions of future of quality assurance combined with personal views and experiences and DevOps maturity level. A draft questionnaire was presented to Comiq management and finalization of the questions was done in cooperation. Comiq management accepted the final version before official publishing. The questionnaire was conducted in Finnish and the final version is presented in Appendix 1.

The online questionnaire was addressed and sent by email to a group of people who work in relation to quality assurance like testing and quality assurance experts or consultants, test managers, test leaders or people who are responsible for the development of quality assurance. All participants worked for companies that are Comiq's customers or worked for Comiq directly.

5 Implementation

The study was implemented by using a browser based survey tool kyselynetti.com. The tool allows users to create diverse online surveys. The tool required the executer of the survey to register and the service was offered free of charge to Haaga-Helia students. Participating in the questionnaire was allowed without registration.

The tool allowed a self-explanatory interface to build questionnaires quickly. The tool gave the option to choose from a variety of different question types, phrase your own questions, add explanatory text, upload images and thereby systematically design a survey piece by piece. The tool allowed the executor to preview single questions as well as the entire questionnaire. The tool also did not have any time limitations. (Kyselynetti 2017).

The tool was chosen because the user interface was user friendly, easy to use and analysis of results was easy.

The questionnaire was sent to approximately 350 people by email. It was addressed to a select group of Comiq customers and personnel during 22.6.2017 – 30.6.2017 and the questionnaire was held open until 7.7.2017. During the time the questionnaire was open, a total of 93 people participated. The questionnaire had a total of 15 questions, four of which were open questions.

The resulting analysis was performed by importing all the data to Microsoft excel worksheets and the creation of figures was done with the help of excel graphics and features.

6 Analysis

In this chapter, the results of the questionnaire are presented in more detail.

6.1 Background of participants

Question 1: Current job description.

Type of the question: Single selection with optional text field (mandatory).

Total number of participants: 93.

Figure 7 shows the participant's current job title distributions.

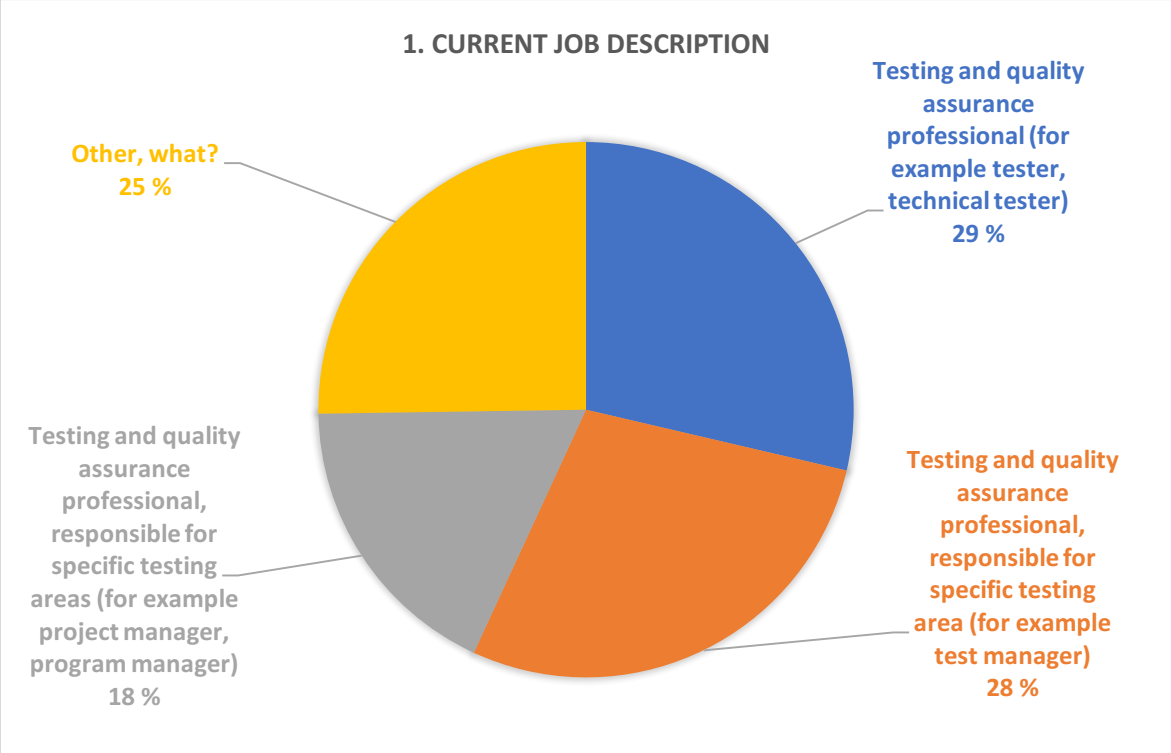


Figure 7. Answer pertaining to question 1 as a percentage, all participants

24 participants (25%) answered that the current job description is something other than the given options. 10 participants current job description was directly related to quality assurance. Most of them are currently responsible for overall quality assurance or work at management level. 14 participants' current job descriptions were related to IT departments, information management, system architecture and developing future systems or similar.

Question 2: Do you work for a company that specialises in testing and quality assurance?

Type of the question: Yes / No –question (mandatory).

Total number of participants: 93.

Figure 8 shows the results to the question of whether the answerers are working for a company that is specialised in testing and quality assurance.

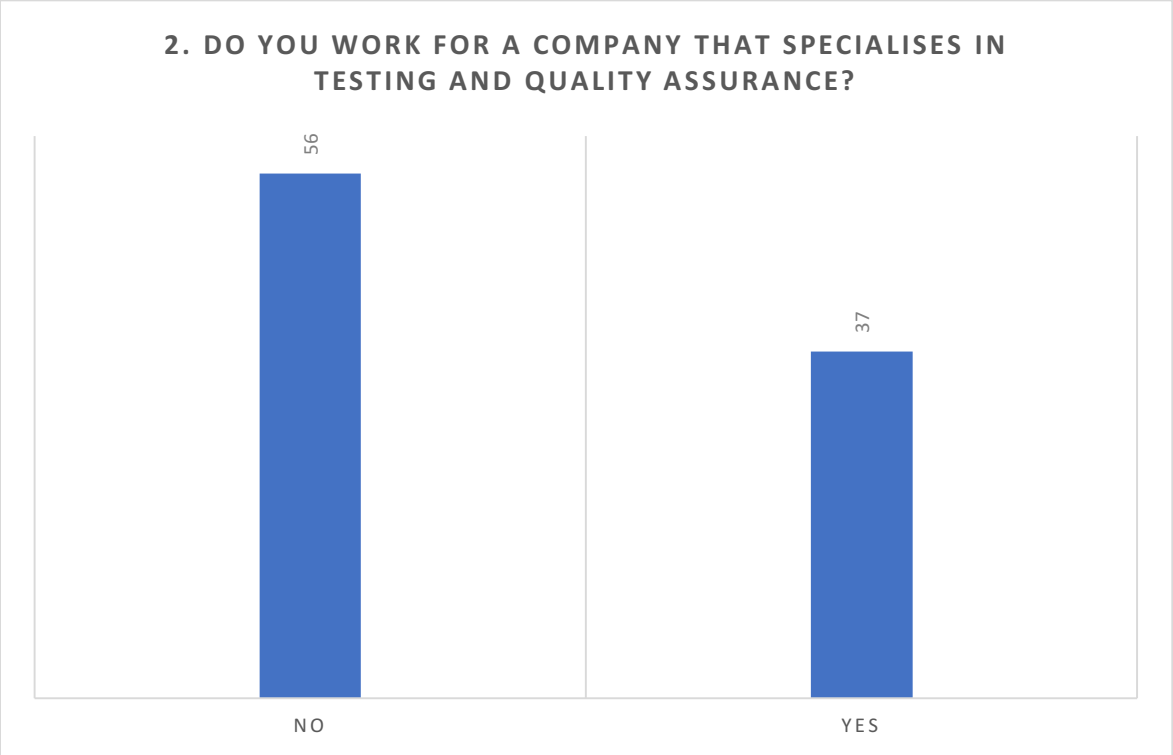


Figure 8. Answer pertaining to question 2, all participants

Question 3. How many years have you worked in testing and quality assurance?

Type of the question: Single selection, listed (mandatory).

Total number of participants: 93.

Figure 9 shows the participants years of experience in testing and quality assurance.

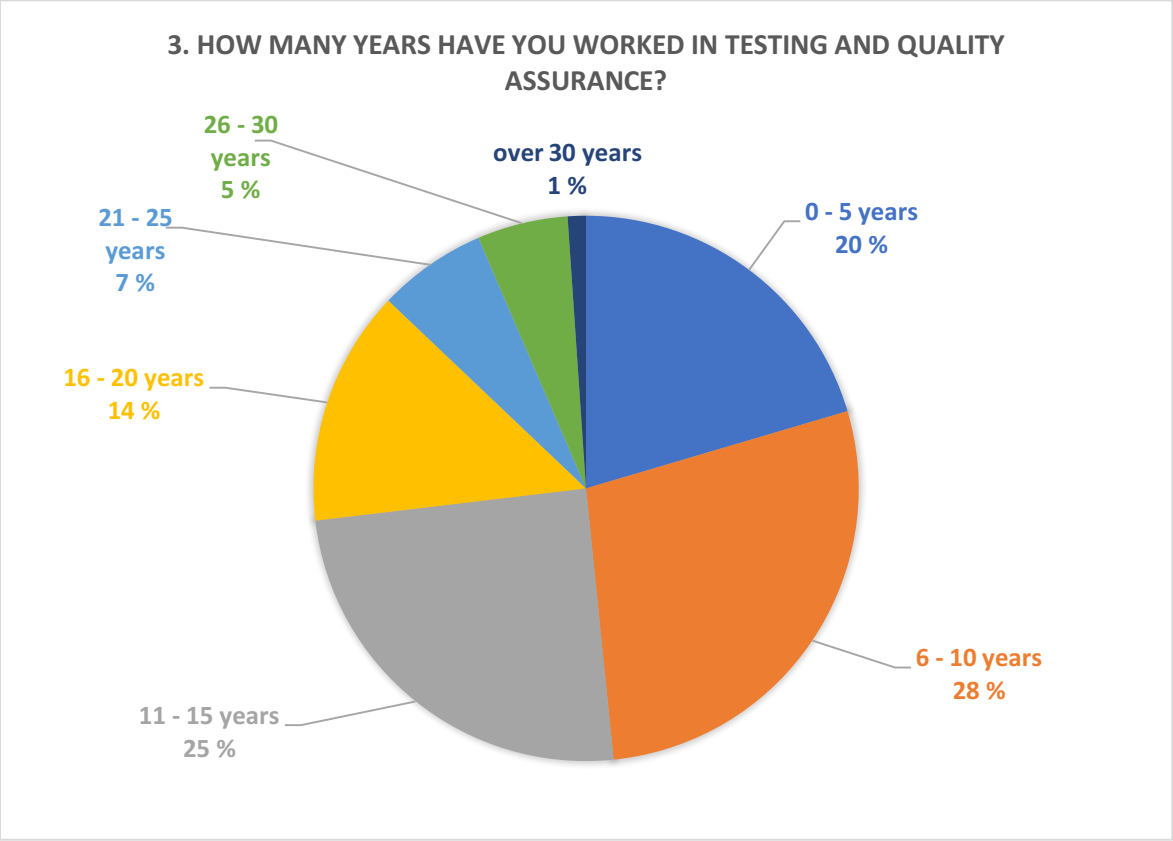


Figure 9. Answer pertaining to question 3 as a percentage, all participants

6.2 Properties of quality assurance experts in the future

Question 4: In the following has been listed typical properties for testing and quality assurance expert. How important do you see the following properties?
Type of question: Evaluation table (mandatory).
Options for each property: Minor (1), quite minor (2), important (3), very important (4), don't know. "I don't know" answers (total of 4) are not averaged.
Total number of participants: 93.
Figure 10 shows the important properties for testing and quality assurance experts in the order of importance between all participants. Figure 11 shows the answers to the same question and the differences between participant's groups.

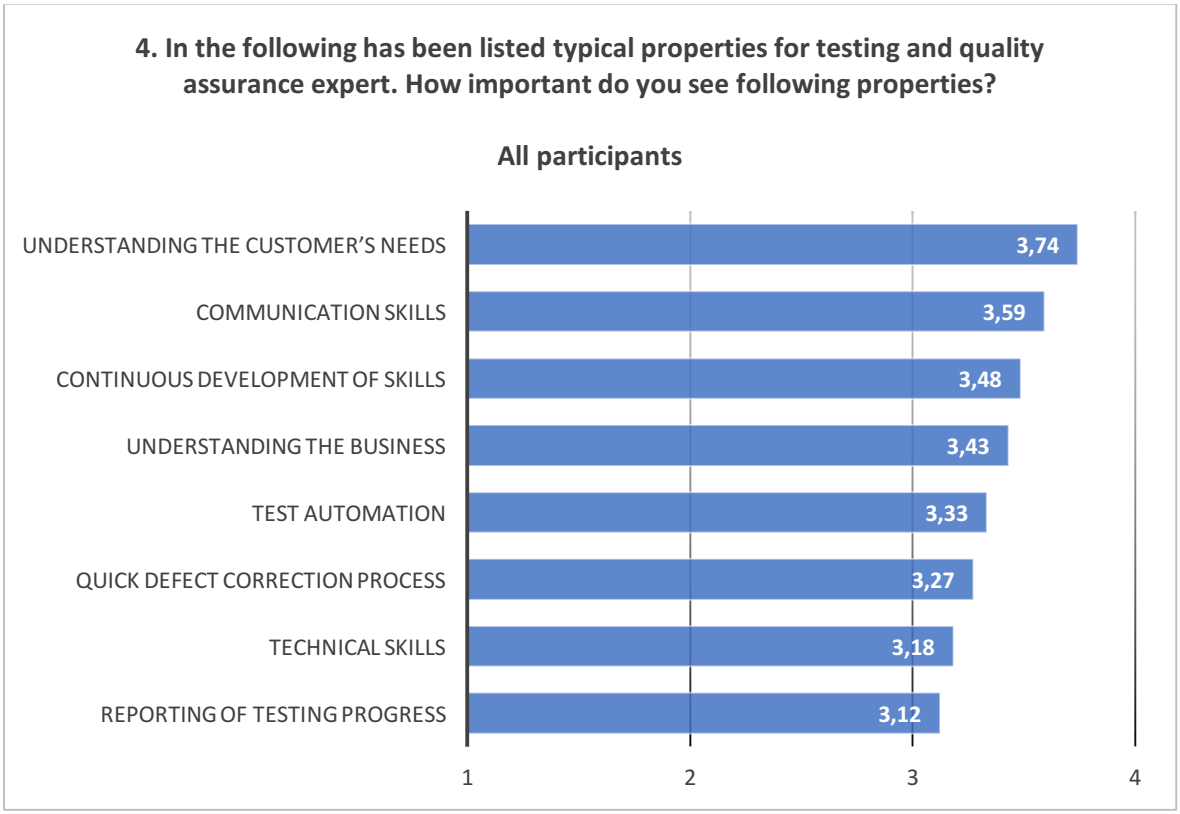


Figure 10. Answer pertaining to question 4 as an average, all participants

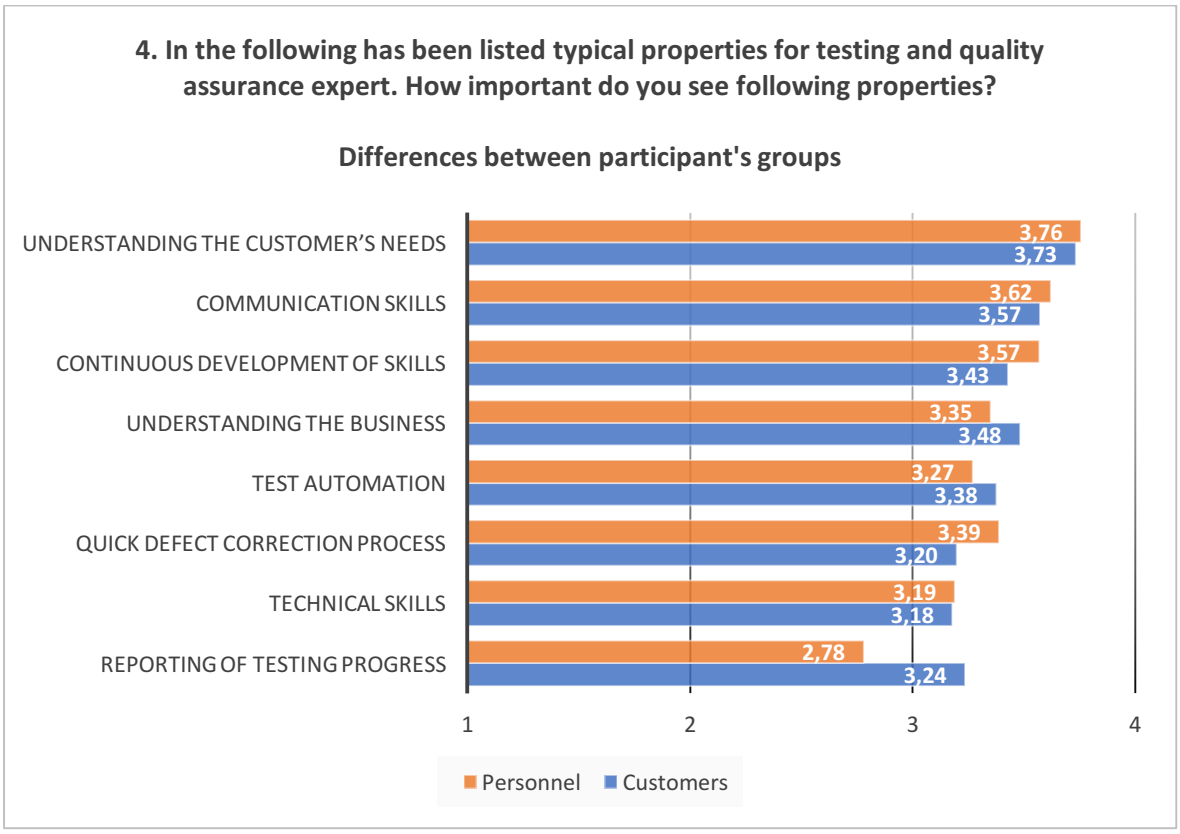


Figure 11. Answer pertaining to question 4 as an average, differences between participant's groups

Question 5: What (other) properties do you hope for testing and quality assurance expert in the future?

Type of question: Multiline text box (optional).

Total number of participants: 41.

Table 1 shows the other important properties for a testing and quality assurance expert in the future and how many times each property was mentioned.

Statement	**Time mentioned**
Understanding the overall picture of the business	15
Test automation (understanding what is beneficial to automate)	6
Developing overall testing	6
Understanding different software developing methods (Scrum, DevOps, SAFe, Kanban)	6
Understanding and developing processes	5
Initiative	4
Understanding the technical aspect of the system or systems	4
Exploratory and experimental approach to testing	3
Accountability and profitability	3
Cooperation skills	3

Table 1. Other important properties and skills for a testing expert in the future

In addition, the following skills and properties were mentioned 1-2 times: Skills and understanding of project and change management, different testing methods, sell and enforce workable practices, test case planning, maintenance of test data, integrity of the integration tests, security testing, tool management, user experience of software under test, concrete communication, artificial intelligence, robotics, different environments (test, mobile, mainframe) and service design.

From a testing expert, the most wanted characteristics were accuracy, systematic approach, creativity, curiosity, confidence, active approach, cooperation skills, emotional and situation intelligence, flexibility, understanding the importance of quality, ability to react fast and learn quickly, questioning, finding undefined paths, sharing your skills, creative intelligence, solution-oriented attitude, setting goals that meet customer expectations, genuine desire and enthusiasm for quality assurance and understanding usability aspects.

6.3 Software development from a testing point of view

Question 6: How important do you see listed matters below in the future?
Type of question: Evaluation table (mandatory).
Options for each property: Minor (1), quite minor (2), important (3), very important (4), don't know. "I don't know" answers (total of 21) are not averaged.
Total number of participants: 92.
Figure 12 shows different important matters in the future in the order of importance. Figure 13 shows the answers to the same question and the differences between participant's groups.

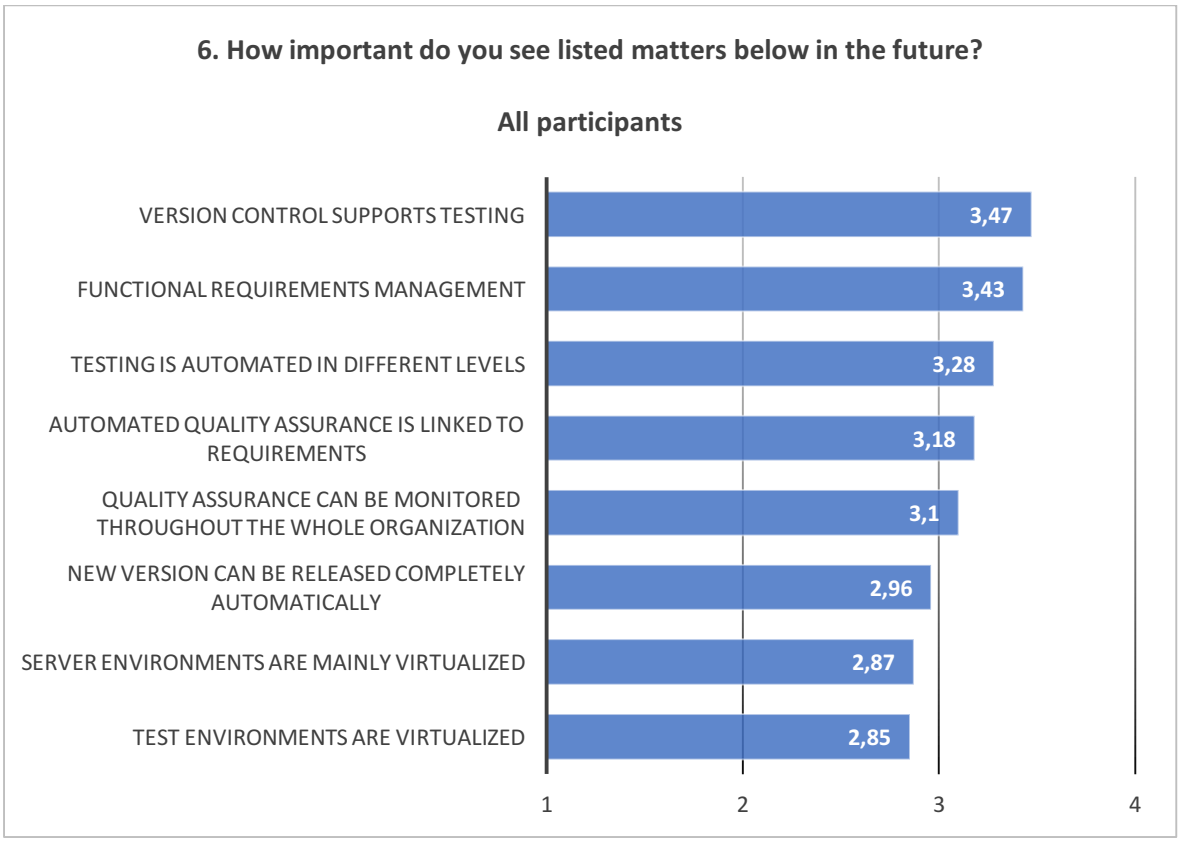


Figure 12. Answer pertaining to question 6 as an average, all participants

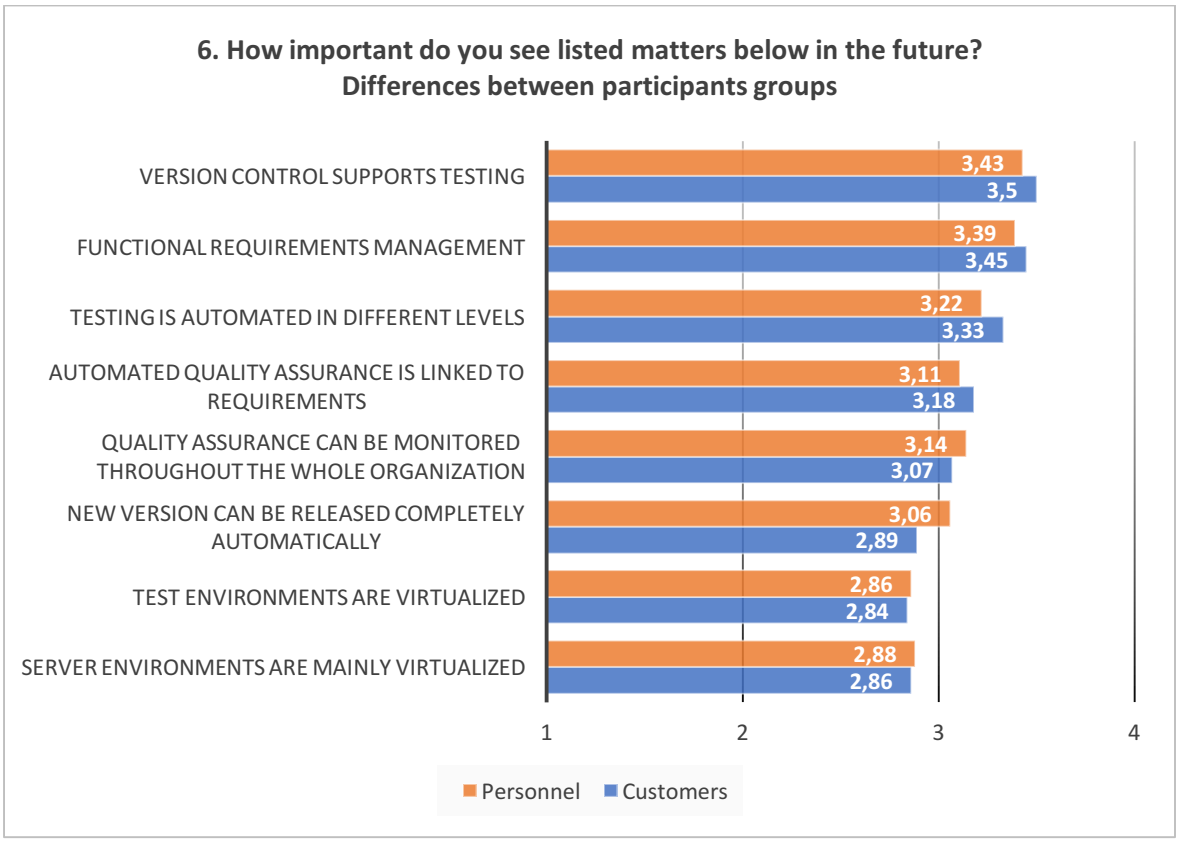


Figure 13. Answer pertaining to question 6 as an average, differences between participant's groups

Question 7: What do you think are the most important matters in terms of quality assurance for software development in the future?
Type of question: Multiline text box (optional).
Total number of participants: 34.

Among the responses, there were six different matters that were mentioned directly or indirectly 45 times. The table below shows the most important statements and how many times each matter was mentioned.

Statement	**Time mentioned**
Overall test automation, including services and environments. Automate only parts that are beneficial.	15
A working development team, cooperation and communication	9
Daily deployment to production is automated	6
Branching and managing different software version and test environments. Acceptance test environment should correspond to production	7
Shift left testing, testing should be initiated right from the start. Transparency of quality assurance	4
Business and customer requirements are in the core of the project. Requirement traceability.	4

Table 2. Other important matters in terms of quality assurance for software development in the future

In addition, the following matters were mentioned either one or twice: overall product quality, security, usability, utilization of machine learning and robotics, control of test data, agile methods, virtualization, engaging the whole organization for testing activities, regression testing, project and program management; well planned and organised development work tasks, artificial intelligence, quality of testing, necessary test report only, physical workplaces (all team members are in the same place), fast defect fixing and risk-based thinking.

6.4 Test automation in the future

Question 8: How important do you see the automation of following test types in the future?
Type of question: Evaluation table (mandatory).
Options for each property: Minor (1), quite minor (2), important (3), very important (4), don't know. "I don't know" answers (total of 5) are not averaged.
Total number of participants: 91.
Figure 14 shows the importance of automation different test types in the future in the order of importance. Figure 15 shows the answers to the same question and the differences between participant's groups.

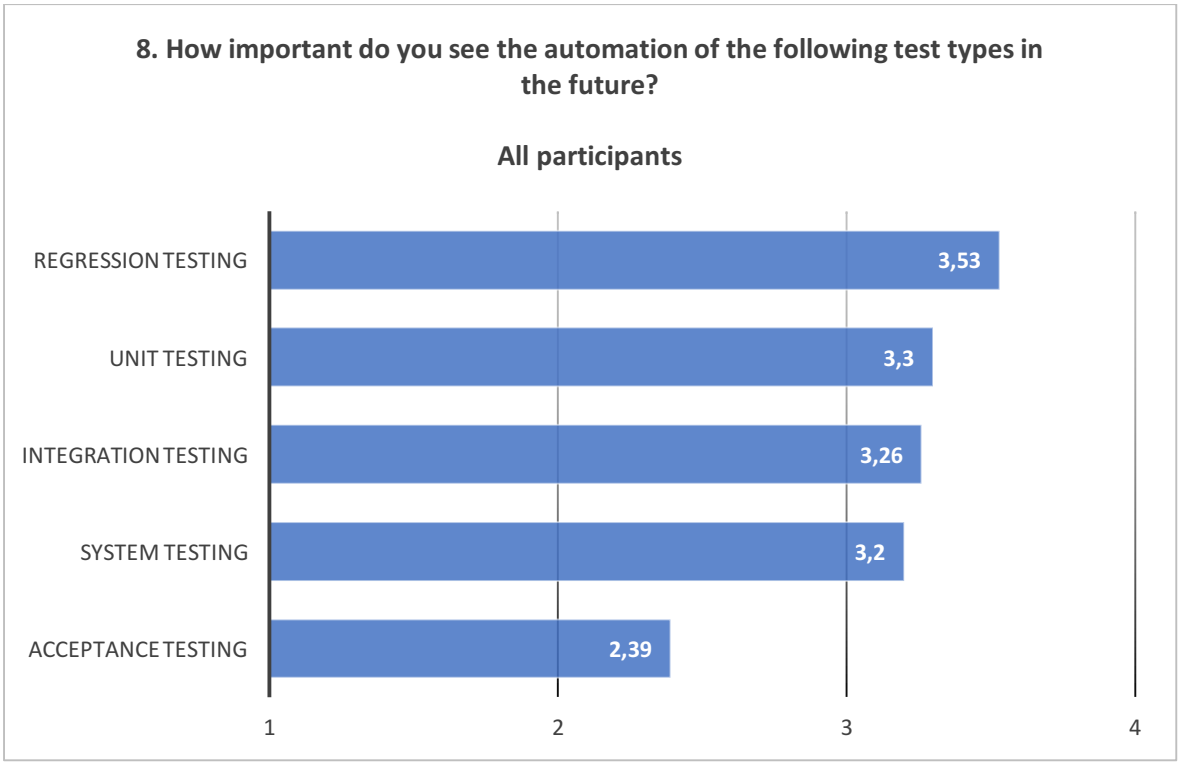


Figure 14. Answer pertaining to question 8 as an average, all participants

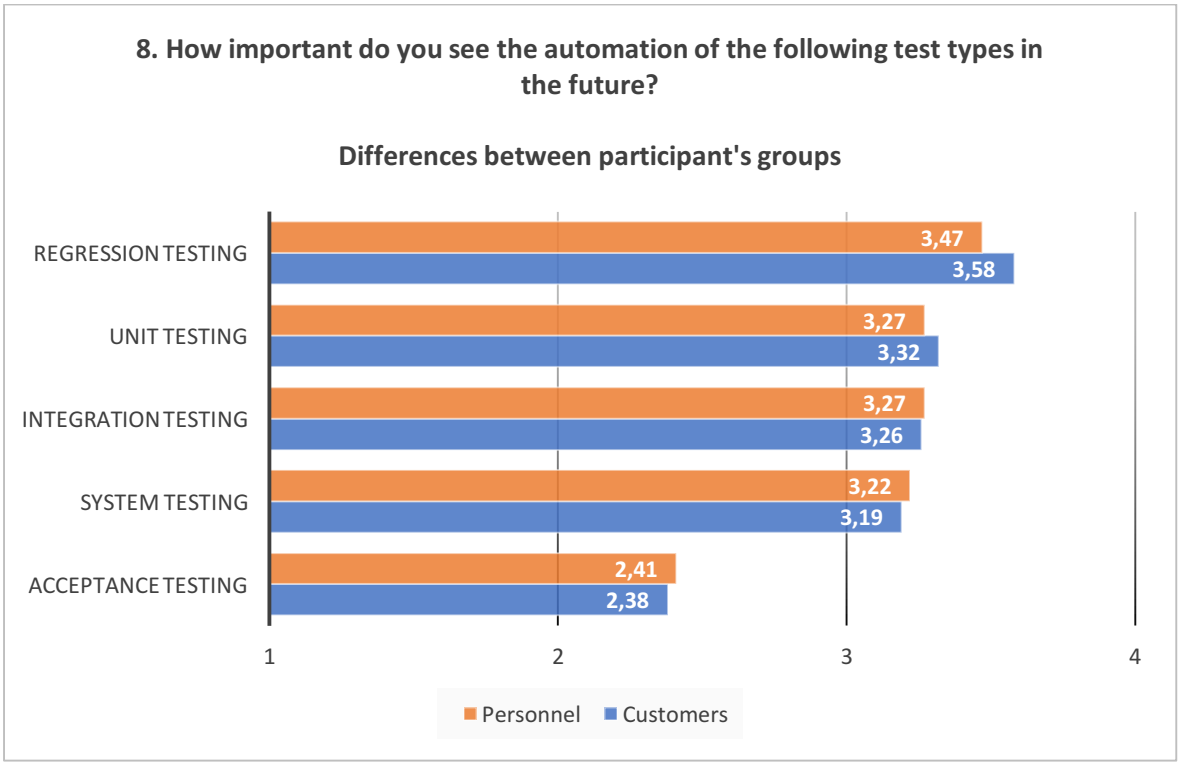


Figure 15. Answer pertaining to question 8 as an average, differences between participant's groups

Question 9: How comprehensive automation should be done in the future?

Type of question: Single selection, listed (mandatory).

Total number of participants: 91.

Options for each property: Not at all, small part, the greater part, all possible and profitable

Figure 16 shows the results on how comprehensive automation should be done in the future. Figure 17 shows the customers answers and figure 18 shows the personnel answers.

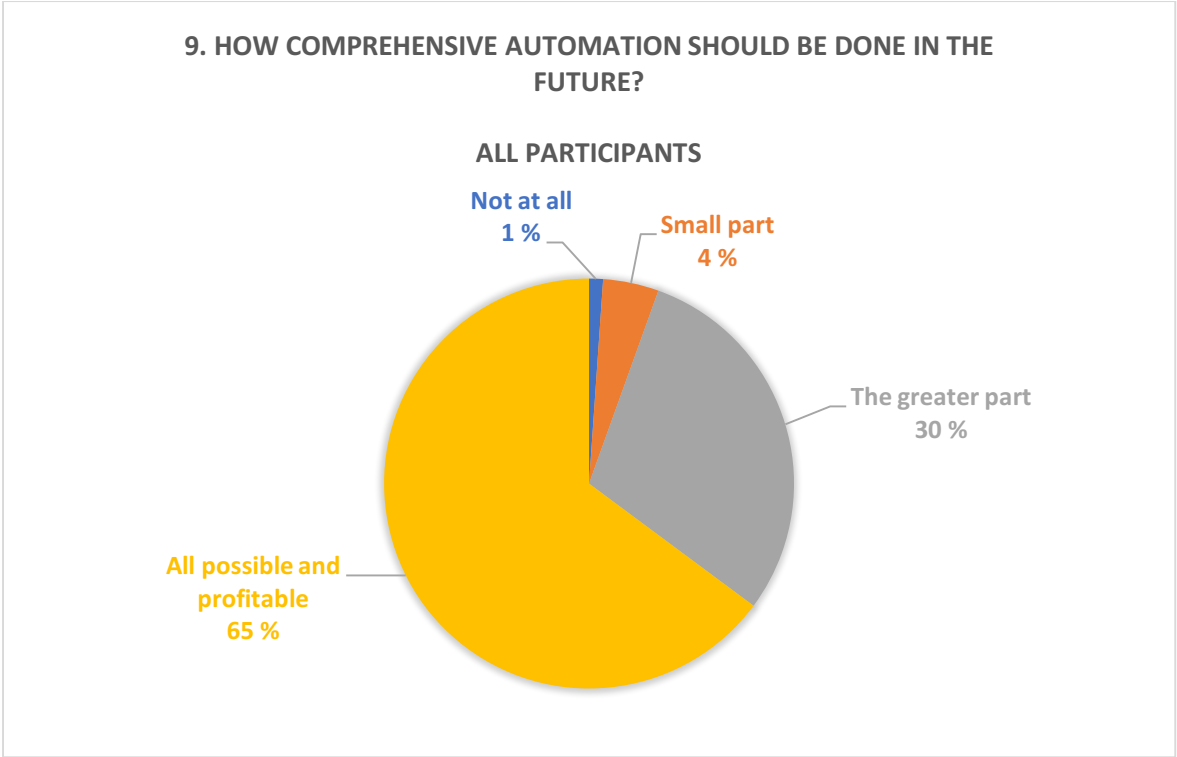


Figure 16. Answer pertaining to question 9 as a percentage, all participants

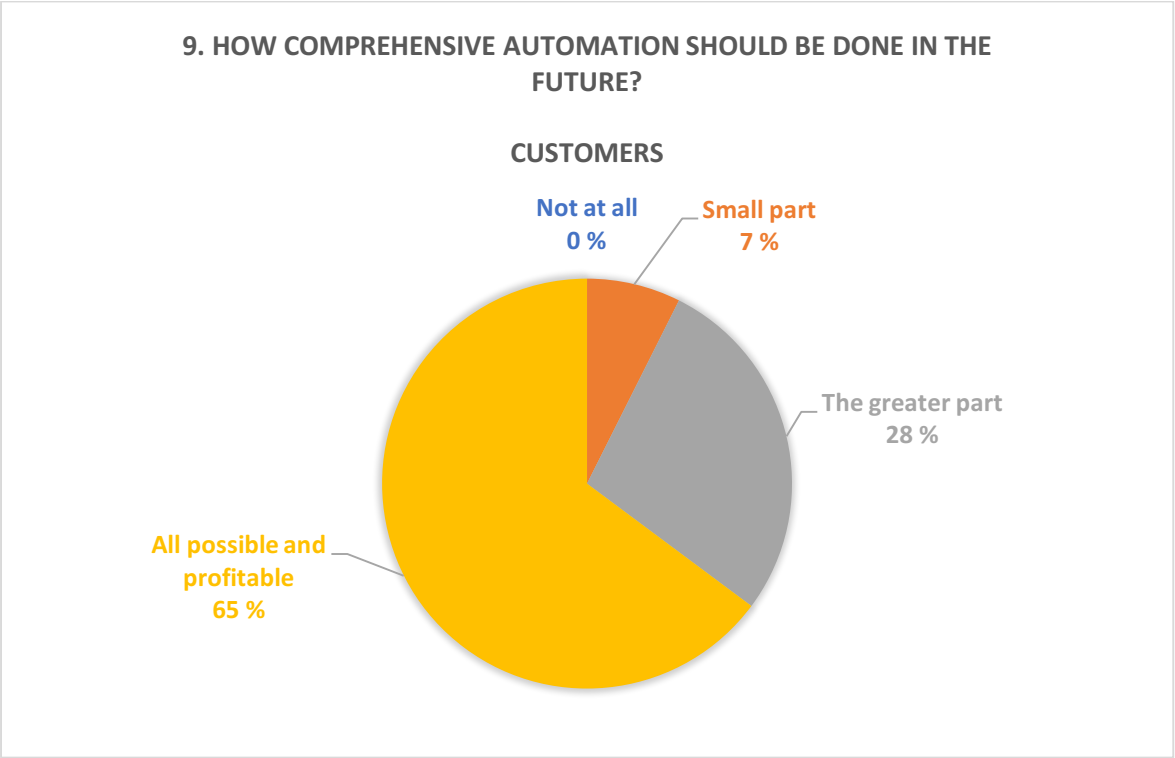


Figure 17. Answer pertaining to question 9 as a percentage, customers

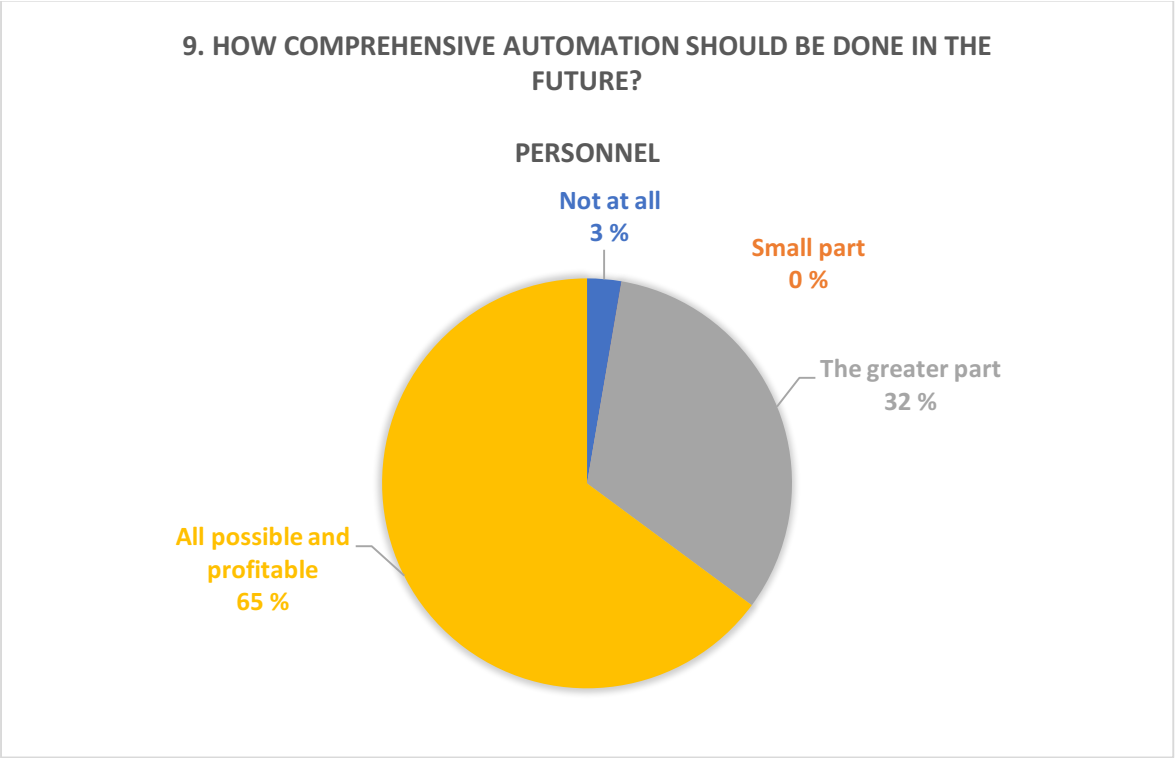


Figure 18. Answer pertaining to question 9 as a percentage, personnel

Question 10: How important do you see the automation of testing environments in the future?

Type of question: Evaluation table (mandatory).

Options for each property: Minor, quite minor, important, very important, don't know.

Total number of participants: 91.

Figure 19 shows the results of importance of automation of testing environments in the future by the percentage. Figure 20 shows the results to the same question and the differences between participant's groups.

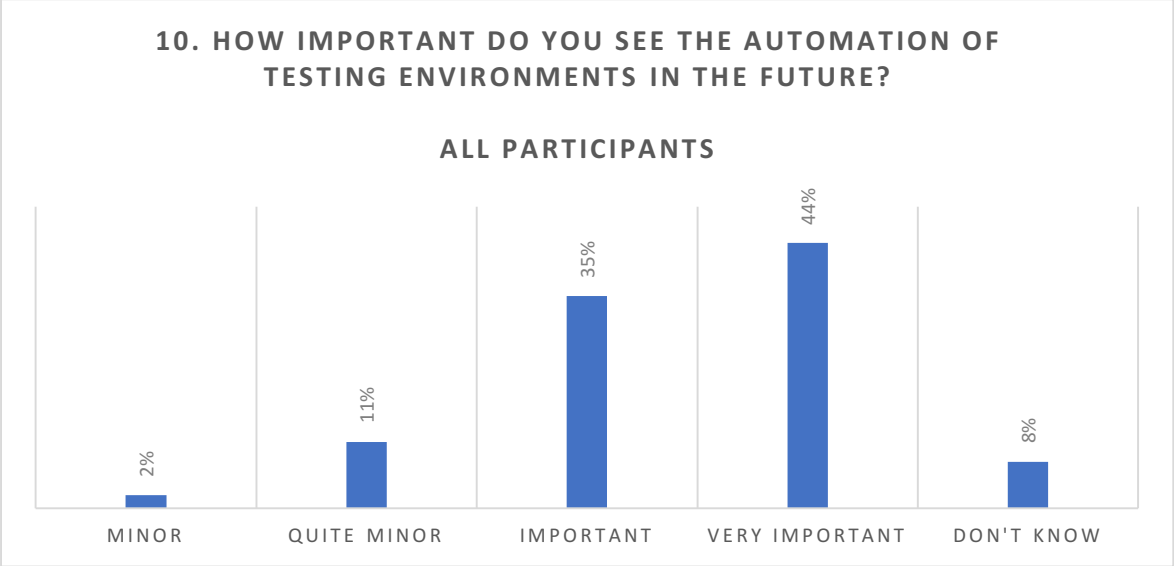


Figure 19. Answer pertaining to question 10 as a percentage, all participants

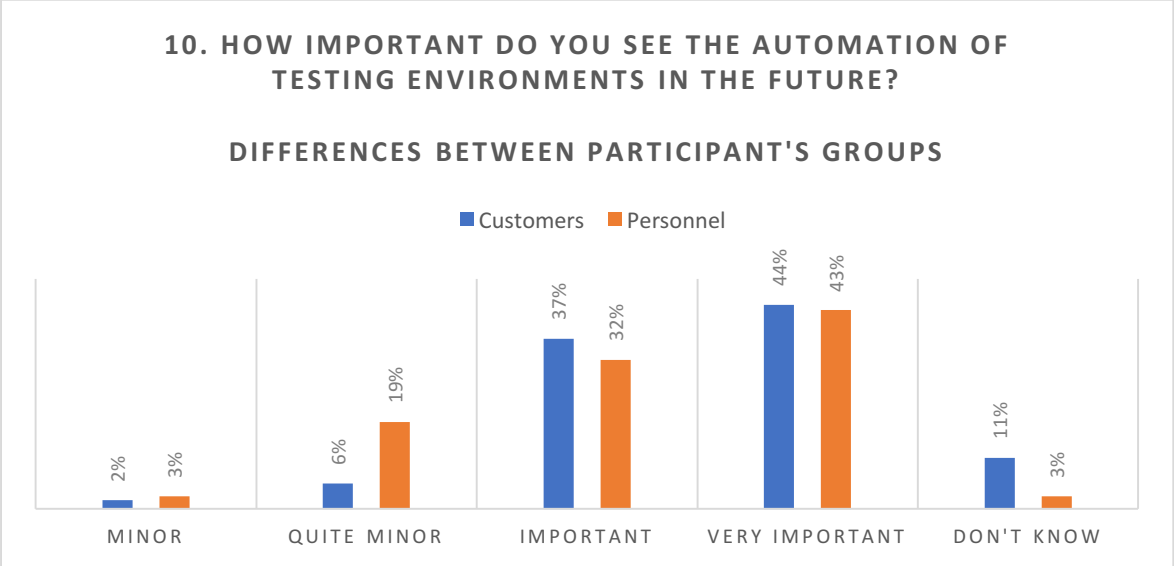


Figure 20. Answer pertaining to question 10 as a percentage, differences between participant's groups

6.5 View of frequency of software versions

Question 11: What do you think is a suitable release cycle for new software versions in the future?

Type of question: Single selection, listed (mandatory).

Options for each property: Daily or more frequent, weekly, monthly, quarterly, semi-annually, yearly.

Total number of participants: 91.

Figure 21 shows the results of suitable release cycle for new software versions in the future by percentage. Figure 22 shows the results to the same question and the differences between participant's groups.

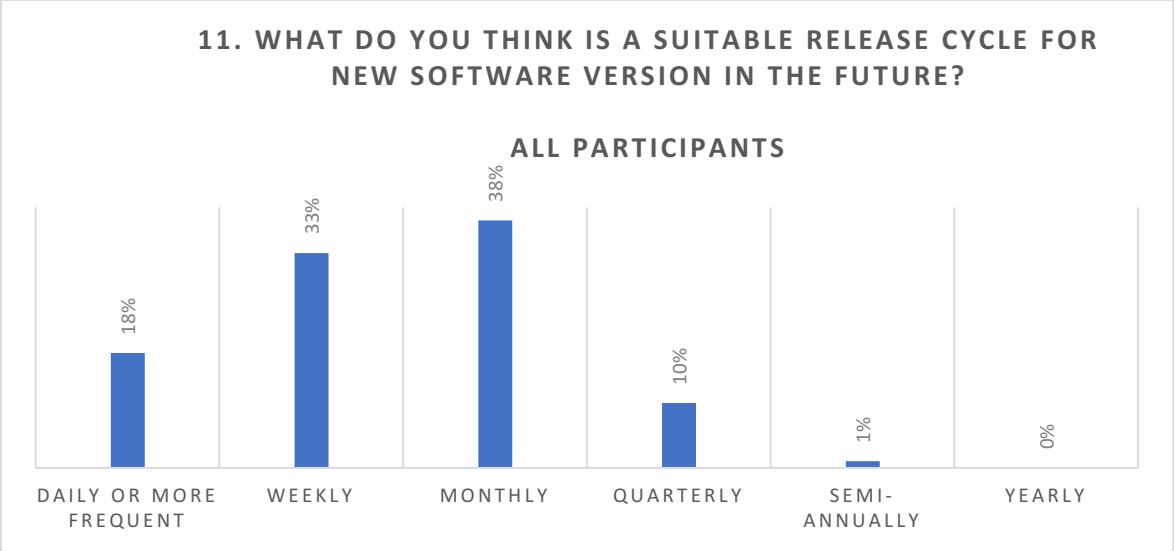


Figure 21. Suitable release cycle in the future as a percentage, all participants

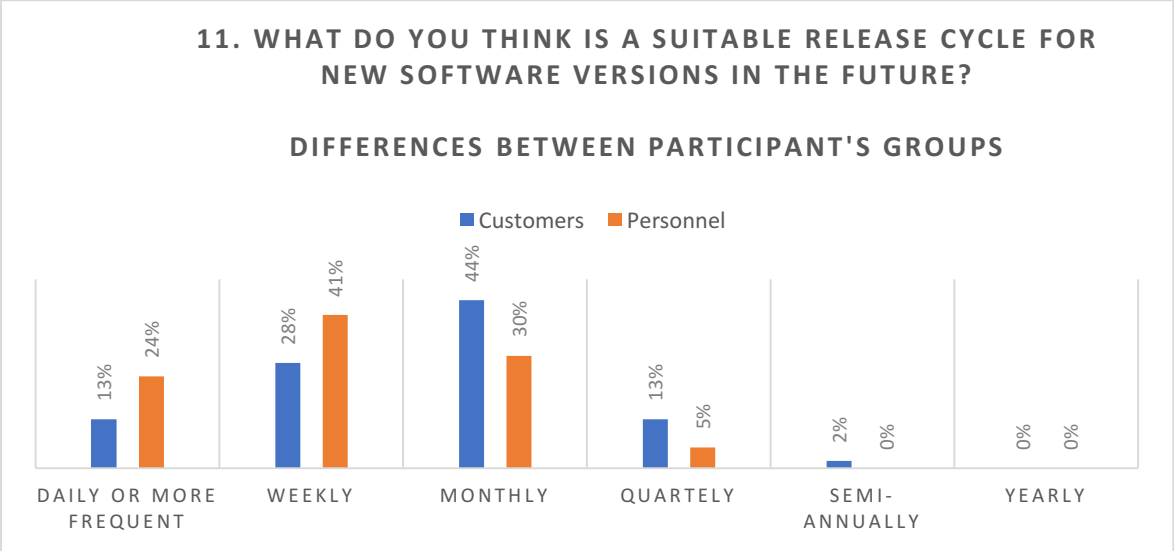


Figure 22. Suitable release cycle in the future as a percentage, differences between participant's groups

Question 12: Do you think the fast release cycle will give you a competitive advantage in the future?

Type of question: Yes / No –question (mandatory).

Total number of participants 91.

Figure 23 shows the results of whether the participants will see a fast release cycle bringing a competitive advantage in the future. Figure 24 shows the customers answers and figure 25 shows the personnel answers.

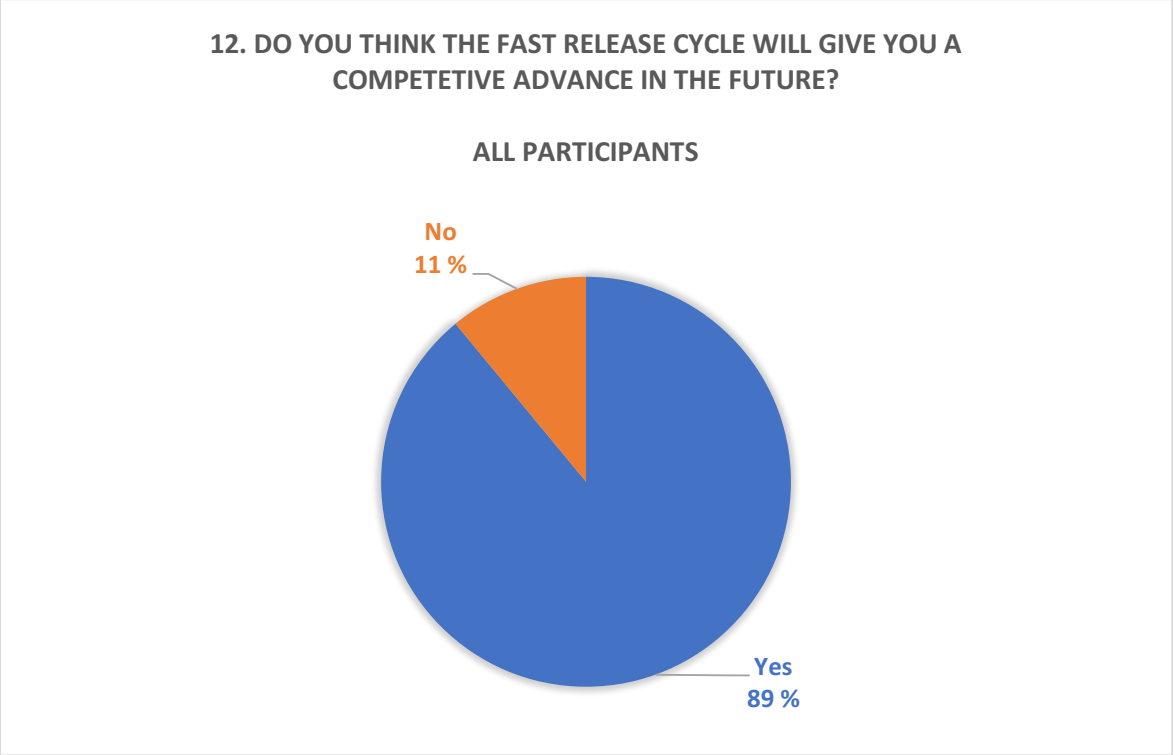


Figure 23. Answer pertaining to question 12 as a percentage, all participants

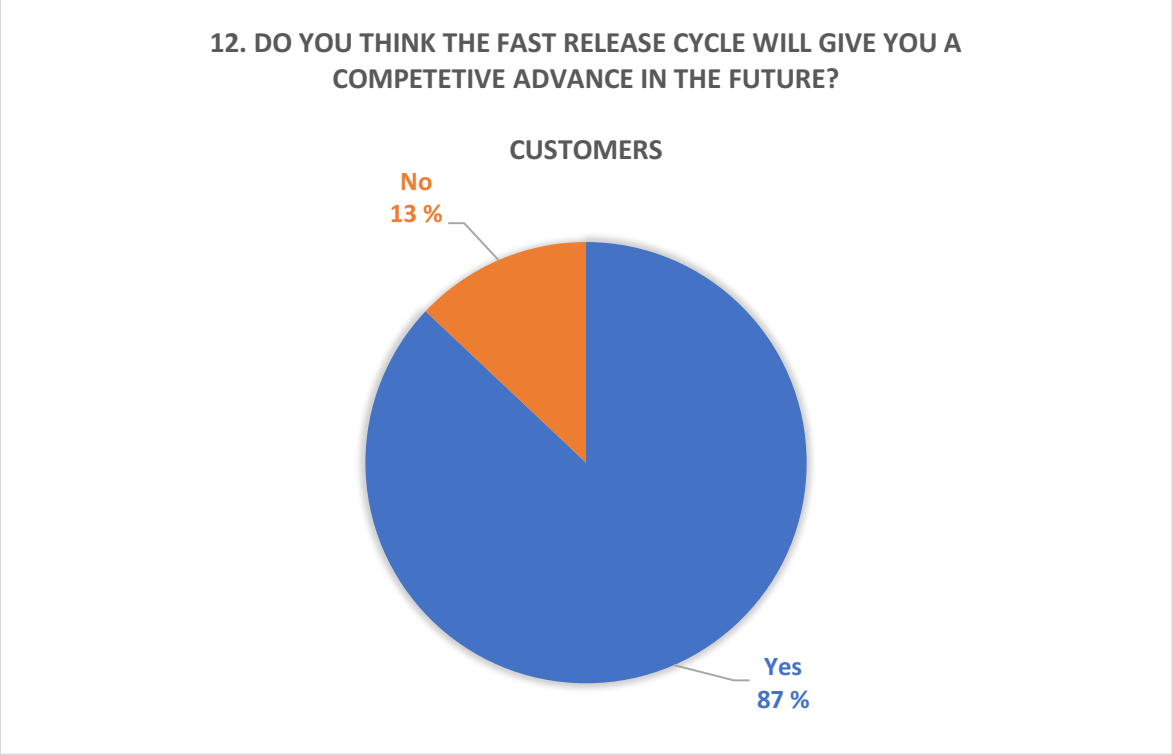


Figure 24. Answer pertaining to question 12 as a percentage, customers

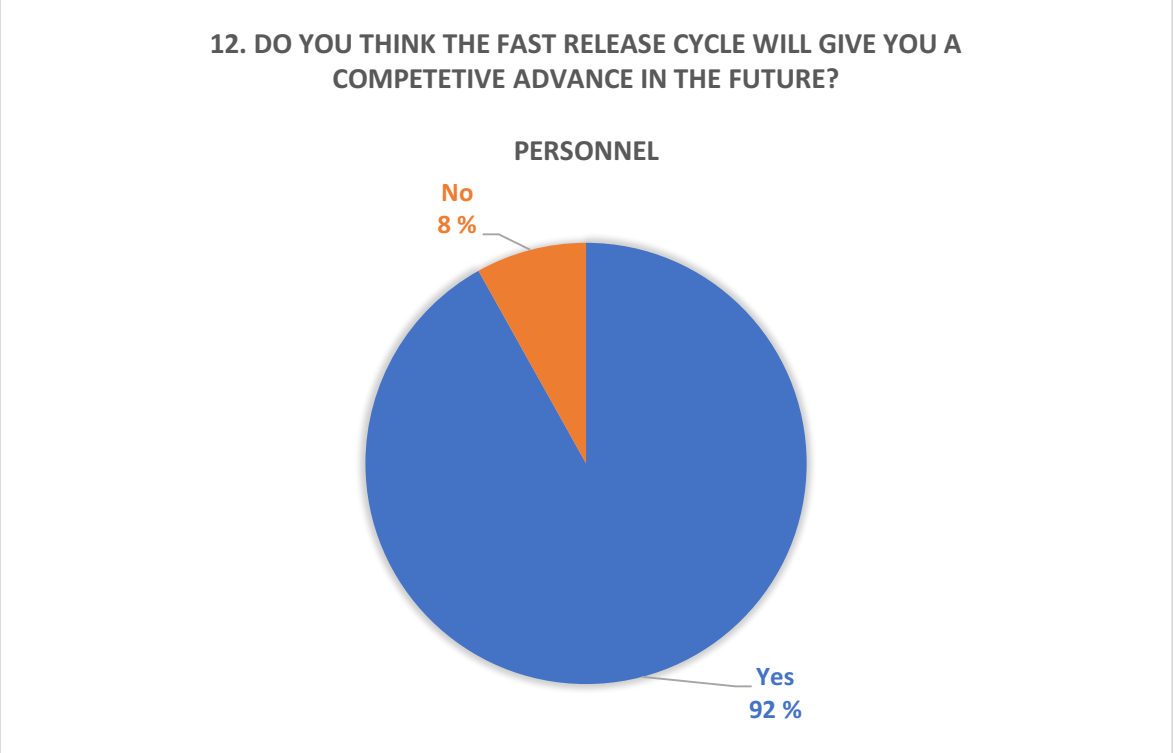


Figure 25. Answer pertaining to question 12 as a percentage, personnel

6.6 Other open questions

Question 13: Based on your own experience, how well does your current state of quality assurance match to your future target state?

Type of question: Visual analogue scale (mandatory).

Options and scale: 0 = Does not match the target state, 100 = Match fully the target state.

Total number of participants: 90.

Figure 26 shows how current state match to the future target state in a scale from 0 to 100 at intervals of 0, 20, 40, 60, 80 and 100. Figure 27 shows the results to the same question and the differences between participant's groups.

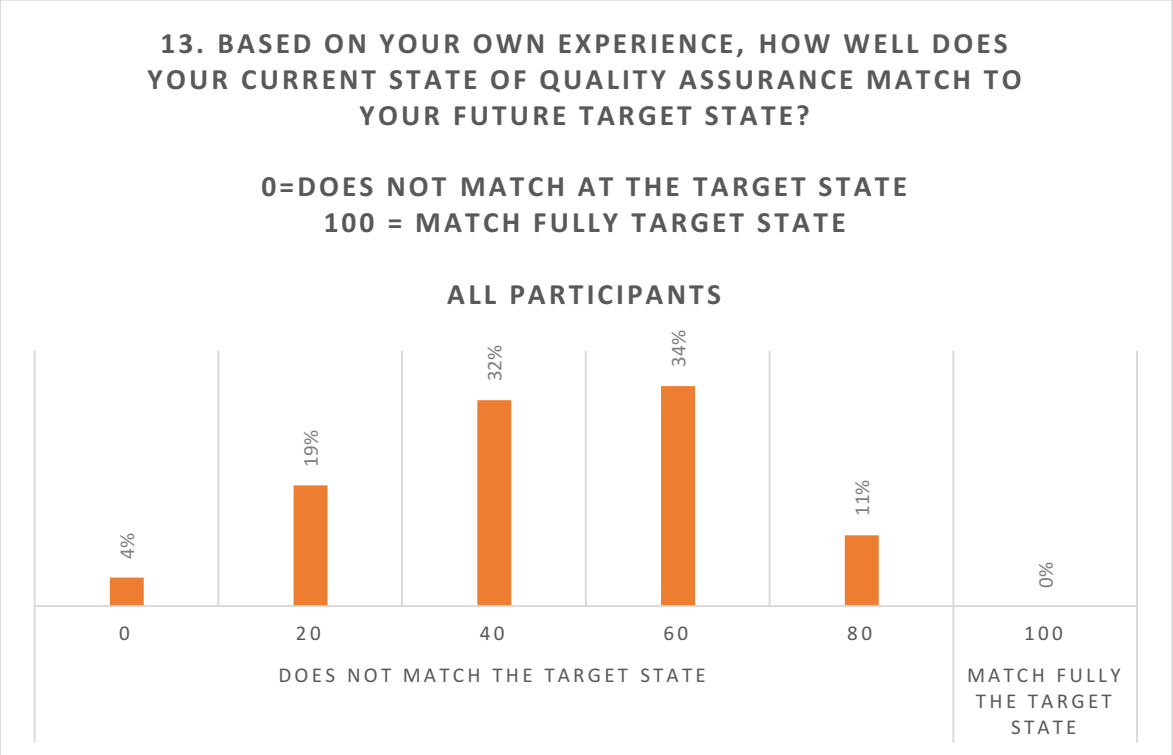


Figure 26. Answer pertaining to question 13 as a percentage, all participants

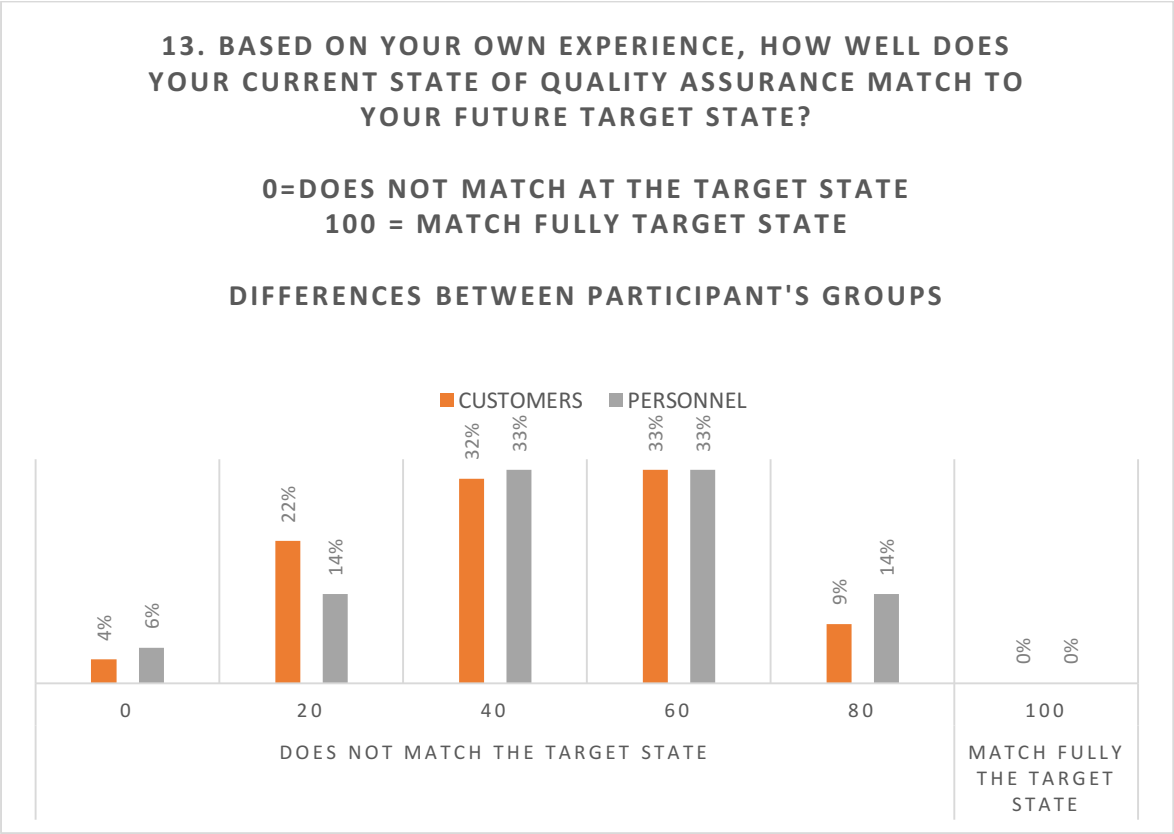


Figure 27. Answer pertaining to question 13 as a percentage, differences between participant's groups

Question 14: Referring to the previous question, what do you think are the biggest obstacles to achieving the target state?
Type of question: Multiline text box (optional).
Total number of participants: 63.

Among the answers there was two main issues that were mentioned over 20 times. Firstly, based on the answers the biggest obstacle was lack of test automation and maintenance of test automation. Test automation was mentioned 26 times. Secondly, the biggest obstacle was related to lack of money, resources and time, mentioned 23 times. The amount of automation should be significantly increased to speed up the exporting of the new features and export them into production. Making a test automation takes resources and resources should be available, especially in the early stages.

The third biggest obstacle was current software development methods and ways of working. Organizations may claim that they use agile or DevOps as software development methods but daily working ways remain more a waterfall model. Testing and quality assurance aspects are not part of projects at the very beginning. Software development methods were directly or indirectly mentioned 13 times.

Issues related to test environments were the fourth biggest obstacle. Test environments are not always available and that will cause direct delays to testing and test environments are too complex. Test environments do not correspond to production environment and testing is not as reliable. Test environments were mentioned ten times.

Many participants mentioned that testing and quality assurance are not appreciated as much as it should be. Due to this lack of understanding, the importance of quality assurance was noticeable among the answers. The true value of quality assurance appears not as visible throughout the organization and due to this, quality assurance is seen as difficult to measure. The importance of quality assurance and closely related matters were mentioned seven times.

Creating and maintenance of test data was reported as an obstacle. Continuously changing software versions might cause extra work for test data maintenance, especially when testing the object is a large task involving complex system that are integrated to various associated systems. Maintenance of valid test data is essential for successful testing. In addition, lack of (technical) testing skills and competence was seen as an obstacle and a challenge. Both, test data and lack of competence were mentioned as an obstacle six times.

Other obstacles listed were version control, management and traceability, working conditions related to physical locations, challenges inside teams, lack of manual testing, changing requirements, complex interdependencies between systems, quick releases, last minute changes, old infrastructure, stiffness in organizations, lack of relevant information and that current testing tools don’t support testing as efficiently as required.

Question 15: What do you think is the most important measure to improve quality assurance in the future?
Type of question: Multiline text box (optional).
Total number of participants: 64.

Improving extensive test automation in various levels was by far the most important measure to improve quality assurance in the future. Almost every third answer was related to test automation. The general opinion was that automating all possible repetitive work will release time for as an example very important exploratory testing. Test automation should start from unit testing level and unit tests should be as comprehensive as possible. Automating generated test data was also seen as a benefit. It would bring speed and

reliability to overall quality assurance. An essential aspect of test automation is the understanding of customer's needs and requirements. In the ideal scenario, the slightest changes could be tested by automation comprehensively on the same day, the changes been made and release would be ready to take in production. In addition, test environment automation as well as continuous integration (CI) should also be part of the automation process.

The second most important measure to improve quality assurance was understanding that testing and quality assurance is an integral part of business. Testing should be appreciated across the whole organization, highlighting that investing in testing right from the start is essential. Projects should always have sufficient expertise to promote quality assurance tasks and benefits should be agreed at management level. Spreading understanding of the importance of testing and quality through concrete examples is an important task. All organizations should also have enough competent people to oversee overall testing and quality, such as test leads, test managers and test coordinators and test experts should always make sure that quality goals are followed and finally achieved.

"Let's start with the basics: The valuation of the test is very high when management is asked. In practice, missing resources are often sourced from 3rd world countries. This will lead to new continuous training circles. Testing professional is testing professional and companies should seek professionals, not savings".

The third most important measure to improve quality assurance was involving testing in to the planning and development right from the start of a new project. Testing should be an essential part of genuine agile teams.

The biggest benefit will be gained by investing in quality assurance as early as possible in the following actions and tasks: estimate test work realistically, give testers a role to give a feedback of software quality, involve testers to review requirements, and allow testers to help developers as much as possible for example unit and integration testing. It is also important to ensure that the test coverage is as high as possible in all testing levels. In overall, the key for success is close cooperation in quality aspects.

Improving quality requires beneficial cooperation between parties and requires common operating models (such as Lean, Agile and DevOps) and the utilization and competence of quality assurance and testing methods and systems. Collaboration needs to be genuine in the organizations and needs continuous improvement. Also, continuous training and

necessary resources are essential for success. 25% of the answers were related to common models, tested related training and resources.

The rest of the quality improving methods were improving communication, giving developers more responsibility of testing, suitable testing tools, improving transparency, quick and light releases, learning from mistakes, functional development environments, virtualization of testing environments, development needs to be more methodical, minimizing quality debt and importance of exploratory testing.

7 Discussions

This chapter presents the research findings from the viewpoint of the author and refers to the analysis. The purpose of this chapter is to answer the research questions, which are presented in chapter 2 and discuss the findings in more depth.

7.1 Answers to the research questions

1. What views and expectations customers of Comiq have about future trends in the field of quality assurance?

   Customers are expecting testing to be part of the software development right from the start and be an integral part of software development. Cooperation between testers and the whole organization is one of the main keys to success and testing is no longer the last action before launch to production. Overall quality assurance is going to be appreciated more in the future and organizations are ready to invest more in quality. This refers to the answers given to question 15.

   Comiq customer's views and expectations are similar to current predictions. Customers are expecting improvements to effective test automation at many different levels, including automation of test data and automation of test environments. This refers to the answers given to questions 6, 7, 8, 9, 10, 14 and 15.

2. How current trends and development methods will impact in the fields of testing and quality assurance in the future?

   Testing and quality assurance will gain a larger part of overall software development. DevOps maturity will continue to grow and new software versions are going to be released more often because fast release cycle will give competitive advantage. This refers to the answers given to questions 6, 11, 12 and 13.

3. What should be the focus areas of Comiq (from training, studying point of view) in the future to update and maintain employee's high competence levels?

   The most important area to focus on in the future is test automation. The importance of test automation was raised in multiple answers and lack of automation was also seen as one of the biggest obstacles of achieving the target state in quality assurance. Other main areas to focus on are understanding customer's needs, communication

skills and effective ways of working. It is essential to follow current trends and develop employee's skills in DevOps. Developing overall skills was appreciated among the customers and developing skills should be continuous. This refers to the answers given in all the questions.

7.2 Background of participants

The questionnaire was aimed at two main groups, Comiq's customers and Comiq's personnel. From all the participants, 60% were customers and 40% were personnel of Comiq.

Three out of four participants (75%) were directly working in the field of testing and quality assurance either as a tester, technical tester or person responsible for specific testing area. Most participants answering option were related to testing either directly or indirectly. Overall the group of participants can all be described as testing experts and therefore the research reached and an adequate goal-oriented audience.

Most of the participants (73%) have been working in the industry between 0 – 15 years. The rest of the participants (27%) have worked 16 years or more in the industry.

7.3 Properties of quality assurance experts in the future

The results show that all listed features (*understanding the customer's needs, communication skills, continuous development of skills, understanding the business, test automation, quick defect correction process, technical skills and reporting of testing progress*) are all considered as very important, as the average of all answer options are scored over 3 (very important). The four most important outcomes (*understanding the customer's needs, communication skills, continuous development of skills and understanding the business*) are all related more towards an individual's work ethic and personal ability and they are not directly related to purely testing or technical skills. The rest of the outcomes (*test automation, quick defect correction process, technical skills and reporting of testing progress*) are all directly related to testing skills that individuals can improve by experience and or studying.

This result indicates that anyone with the right mindset can grow in the future as a respected expert on testing and quality assurance. Test automation is still seen as important (average 3,33) and this result supports the current trends in software development. In my opinion, developing the skills of test automation would be the sector I would recommend investing for the future.

In the analysis, the differences between the different answerers were small. The biggest difference was that personnel did not see reporting of testing progress as important as customers.

41 participants answered to the open question about other important properties of testing and quality assurance experts should have in the future. The most important property is an understanding of the overall picture of customer's business, software development methods and importance of quality. I believe that the key is to have good overall competence that combines all these skills and understanding the whole business is a mandatory base skill for testers. On top of this skill, testers can grow more detailed competence according to testing and technical related issues and continue to be a testing expert.

The study also proves that personal properties such as initiative, accountability, profitably and cooperation skills are seen crucial for testing experts. These properties can also be improved by time and experience in diverse projects.

7.4 Software development from a testing point of view

In DevOps principles, there are many aspects and parts that will directly help to improve the quality of software. From listed answer options (*version control supports testing, functional requirements management, testing is automated in different levels, automated quality assurance is linked to requirements, quality assurance can be monitored throughout the whole organization, new versions can be released completely automatically, server environments are mainly virtualized and test environments are virtualized*), the first five options were seen as least important or higher because the average score was three or higher.

This result indicates that organizations see the benefit of DevOps principles but not yet completely. Current trends and DevOps principles highlight the importance of automated (new) version releases but this aspect was not seen as important and the average result was 2,96. Related to automated version release, test environments and servers need to

be virtualized and these options were ranked as the least important aspects of current choices. This could also mean that participants are more aware of agile methods than DevOps principles.

34 participants answered to the open question about what is the most important matters in terms of quality assurance for software development in the future. Individual answers were closely related to the options in previous question and in addition participants mentioned a few other features, such as team cooperation, acceptance test environments corresponding to production environment, shift left testing, transparency of quality assurance and requirements traceability. These matters are related to both, agile and DevOps principles and the results indicate that participant's organizations need to understand also core matters to create successful software development methods.

7.5 Test automation in the future

Developing test automation has been one of the top matters for years now and test automation is currently seen as one of the most important factors to improve the quality of software. Test automation implementation is necessary also for various test levels and types. Based on the research results, the most important is to automate regression testing. This result indicates that the focus in software development is usually in implementing and testing new features and therefore automating regression testing saves time testing features that have been implemented and tested in previous phases. This makes perfect sense if the goal is to create new features. If the goal is to develop existing features, for example make software more usable or improve customer expectations, regression testing is not necessary beneficial, depending the core solution of implementation.

The results suggest that test automation is considered important on unit testing, integration testing and system testing levels. Testing at acceptance test level was seen less important, average score was only 2,39. This study demonstrates that this might be an indication that organizations are not aware or do not regularly use Acceptance Test Driven Development (ATDD) as an approach and developing is done at lower levels. Overall maturity of DevOps is not yet reached.

Participants were asked how comprehensive automation should be done in the future. Exactly 65% of all participant's thought that all possible and profitable features should be automated. Another interesting result was everyone on the customer's side thought that automation should be always done, either the greater part (28%) or small part (7%).

Comprehensive automation is clearly seen as an essential part of software development that result is completely in line with current predictions of the future.

Personnel's answers were divided slightly differently, 32% thought that greater part should be automated and 3% thought that automation should not be done at all. In overall result was that 97% of personnel saw comprehensive automation as important.

The third question considering automation was related to the importance of test environment automation. Among all answers 79% considered test environment automation either important or very important. Interesting point in this question was that the customers saw automation environments more important than the personnel group. The difference was not particularly large but it was noticeable; 81% of customers and 75% of personnel. Few participants (customers 11%, personnel 3%) answered that they don't know or have no opinion of test environment automation. Environment automation is a relevant part of DevOps and therefore maturity is not yet fulfilled completely.

7.6 Frequency of software versions and competitive advantage

The question about suitable release cycle for software version suggests that, 89% of participants believe the new software version should be released monthly or faster. 10% answered that a suitable release is quarterly and semi-annually only 1%. In fact, the most common answer was monthly (35%). This indicates that frequent releases are an important part of software development and this result supports the current predictions of speeding up the delivery cycles frequency.

In this question, the differences between participant's groups were relatively significant. Over half of customers answered that the release should be monthly (44%) or quarterly (13%) and personnel answered daily or more frequent (24%) or weekly (41%). This proves that personnel see more frequent releases more suitable in the future.

89% of all participants answered that fast release cycle will give a competitive advantage in the future, however 11% thought that fast release cycle will not give any advantage. There was not a big difference between participating groups, the majority of both groups identifying that fast releases will give an advantage. This result is in line with the previous question and it is obvious that business advantage can be achieved after the system, production or service is in the production, not when it is under development. Therefore, fast releases will give advantage to the organization given that releases are done in a controlled manner and that quality is assured.

7.7 Current and future target state of quality assurance

Participants were asked, based on their own experience how well do they think their current quality assurance match their future target state. The given scale was from 0 to 100, 0 meaning that does not match the target state and 100 meaning that match fully the target state. Interesting result was that no one answered that their current state match fully to the target state. This means that all the participants believe there are ways to improve current state.

The majority of all participants answered either scale 40 (32%) or 60 (34%). This indicates that current state is half way of target stage. I believe this means also that many organizations are aware of new methods and ways to improve quality assurance but goals have not yet been met and therefore there is a long way to go ahead. Comprehensive quality assurance consists of many different areas and requires commitment from the whole organization and all areas require improvements in the future in order to achieve all the goals. This is not an easy task to achieve and improvements must be implemented.

7.8 Obstacles of achieving the target state

The previous chapter analyzed the level of achieving the target state. It was shown that the target state was not reached and the next question asked what were the biggest obstacles of achieving the target state. The biggest obstacle reported was the lack of test automation.

Current trends predict the growth of automation testing and the result shows that organizations understand the importance of test automation development. Benefits of test automation are understood but the second biggest obstacle of implementing automation is either money, time and resources. This is an interesting point because the point of successful test automation is to improve quality and save money, time and recourses. The key question is how to combine all the pieces and make test automation a beneficial part of overall quality assurance. This leads directly to the third biggest obstacle which is current software development methods and ways of working. Benefits of agile are well known or at least organizations believe that they have embraced the principles but practical methods can be something completely different. Current trends predict growth of DevOps but it seems like organizations have not yet adopted an advanced maturity level.

Another obstacle was lack of appreciation of quality assurance. Study shows that there are still some misunderstandings when it comes to overall quality assurance related

attitudes. The true value of quality assurance is not seen throughout the whole organization and it is not easy to measure actual development tasks. I believe that every organization should have their own individual specified ways of measuring the quality. Measuring quality will give more aspects and give more transparency.

Rest of the obstacles were related to maintenance of test data, lack of technical skills, version control, traceability, team collaboration issues, lack of manual testing, changing requirements, complex interdependencies between systems, quick releases, last minute changes, old infra, stiffness in organizations, lack of relevant information and that current testing tools don't support testing as efficiently as needed. These obstacles can be solved by reviewing organization-specific processes and increasing training. Organizations should regularly inspect these aspects and combine aspects to software development processes. The list of obstacles was relatively long but it does not mean that all the organizations have all the listed obstacles currently affecting their processes and therefore solving one or two obstacles could be enough to achieve the target state.

7.9 The most important measure to improve quality assurance

The goal of the previous question was to find out the biggest obstacles in achieving the target state. Following on to the previous question, the aim of the study was to find out what are the main measures to improve quality assurance. This last question was related to all the previous questions and the goal was to get results of an overall opinion on what is the most important measure. Based on the answers, test automation at various levels was seen the most important measure. Test automation can be taken onto the next levels by generating required test data, environments and releases to the production. This all proves that organizations are more than ready to adopt principles of DevOps and the future will lead more towards automation and continuous integration. The key point behind all the processes is the understanding of the customer's needs. Without the knowledge and understanding all the development work is useless. This indicates that the customer needs must be clear before any action is taken towards implementing improvements.

Another important measure is to involve testing right from the start. Shift left principle is interpretable from study results and it is clearly noticeable that quality assurance should be integral part of the whole software development process. This is a subject that has been under discussion for many years but this is still seen as a challenge or part that is not adopted well enough. This is an interesting point why it often happens that testing is left to the last minute during the development cycle.

The study results show that improving quality requires cooperation between parties and common operating models like Lean, Agile and DevOps. I believe that organizations need these both to be successful in cooperation. Knowledge of methods and good communication skills are one main key to success and both will support quality assurance aspects.

As a summary of final research question, improving quality assurance in the future requires automation, understanding the customer's needs, involving testing early, effective development methods and communication.

7.10 Validity and reliability

This study was done by analyzing results that were collected by online questionnaire. The questionnaire was open only for two and a half weeks. The link to the questionnaire was sent only to a specific group of people so it is very unlikely that someone outside this specified group answered the questionnaire. Most of the participants left their contact details (email) at the end of the questionnaire which indicates that only invited people participated.

Kyselynetti.com –tool created graphs but those could not be used directly because the questionnaire was conducted in Finnish. The tool gave the ability to transfer the data to Excel, where all the graphs were created. All the end results match fully to the kyselynetti.com –tool so it is unlikely that any calculation errors appear in the graphs.

Open questions were the most challenging parts to analyse and all the results presented in this study are the authors own interpretation. Analysis of open questions was where most of the analysis time was spent. It must be noted that most of the questions were related to individuals own experience or opinion so there is no right or wrong answers.

7.11 Recommendations

Based on the research, a few key points emerged clearly through various issues. The main points to be paid more attention to in the future are the understanding of customer needs, the development of test automation and the efficient use of different software development methods, such as agile and DevOps.

As recommendations for Comiq I propose to consider ways of ensuring that employees have access to needed information of current customer's needs at the beginning of each project. This can be done in multiple ways, depending on individual customer, for example

arranging workshops that are planned in cooperation with Comiq and customer. Topics of the workshop can include a walkthrough of the customer's business, business goals and expectations of employees. This matter should be discussed as early as possible, at the latest during the negotiation of an individual's consultant contract. I believe the time invested in consultants training will pay back in short and long term and give good foundation for the cooperation.

DevOps in clearly one of the main talking points when it comes to software development methods. It seems like most customers know about it but maturity is not yet in the target state. Comiq offers DevOps training to own personnel and training could be extended to cover DevOps practices for Comiq customers. Recommendation is to execute tailored DevOps maturity checks for Comiq customers. Main goal should be evaluating current maturity and find measures to improve maturity yearly. Maturity checks should also a have part where discussion concentrates on finding the main factors that are currently slowing the process and adaptation.

By finding the right measures Comiq customers can find more effective ways to implement test automation at various levels and speed up the release cycle. It is shown in the results that most customers believe that quick releases will bring competitive advantage towards their business. Quality assurance can support quick releases if related tasks are executed in a suitable manner.

Another important factor for improving quality was communication and cooperation. The third recommendation is related to communication skills practices arranged by Comiq. Communication can often be considered self-evident part of software development although it is an essential part of the whole process. Importance of communication skills, especially when discussing quality matters should never be underestimated. Each organization has a lot of individuals that have different kinds of communication skills naturally but I believe that communication skills can be improved and practiced in the same way as other subjects, such as technical skills. Communication skills could be improved through the participation of team building exercises.

The importance of automation will only increase in the coming decades, and therefore, I consider it to be particularly important to develop technical skills to support automation. This could be done for example by setting up an individual's training goals during the periods of one and three years over the near future. This discussion could take place during the employee's yearly appraisal. The discussion could cover a practical plan on how to best utilize the yearly training days that are allocated to each employee.

Investing in test automation experts can bring significant benefit to the organization now and in long run. In the future, automating tests is going to be as important as automating test data and test environments so it is crucial to focus on test automation at all possible levels.

Depending on the size of organization I also recommend that all organizations should have a person or team in charge of overall quality matters including test strategy. This person or team can concentrate of finding suitable ways to measure quality and making quality benefits more visible throughout the whole organization. Measuring quality will offer significant information about the quality areas organizations should concentrate on in more detail.

As a graduate research proposal, I will recommend researches related how to help organizations to improve maturity from agile more towards DevOps. In addition, testing and automation tools are being developed to meet customer needs, and tools are constantly being developed. Choosing the right tools will support the organization's business and therefore I recommend doing further research on available tools and their suitability for different types of organizations.

8 Conclusions

Looking at the changes brought about digitalization over the last few decades, it can be said that many of the earlier manual actions are now being performed by automation. For instance, most people now days book their flights and holidays via online services. In Finland, almost all bank and other finance services are available online. Internet of Things is everywhere. The digital revolution has been and will continue effecting our lives. Automation has already revolutionized our everyday life and I believe that development of automation will continue for a long time.

In everyday life, we may not even always notice the matters brought about digitalization because people adopt the changes quickly. Perfectly working services might be unnoticeable but if a certain (critical) service stops working for the end user in production, it is likely that issue is raised as headline news. The importance of quality assurance often become evident when issues arise and a service is unavailable.

More often services are exposed to direct customer interaction and it is currently predicted that testing and quality assurance are going towards business goals such as decreasing time-to-market and customer satisfaction. Based on current predictions and research results, it can be stated that testing and quality assurance will be significantly changed in the future.

Digitalization and its effects is a subject that is under continuous discussion within the information technology sector. I believe that quality assurance is a significant part of overall digitalization and although we talk about technology related matters, the key to success in quality assurance matters is proper communication between all parties. Whilst creating digital services, we always work with people, not with machines.

One of the prerequisites for success is that testing is involved in projects from the start. Testing should be a close part of the development teams from the beginning to the end of the life cycle and should be adapted as an essential part of DevOps. I also believe that principles of DevOps will keep continue to expand further and the future might bring more new aspects to quality matters.

Future quality assurance specialists will understand both sides, customer’s needs and goals as well as technical solutions and the many opportunities test automation at various levels will bring. Future quality assurance experts will have an open mind and are willing to keep their competence level high and develop their selves regularly.

In the future quality assurance experts will understand all the current and arising quality aspects in DevOps and has up-to-date information about the current tools available. It is predicted that organizations quality assurance will gain a larger amount of the overall information technology budget in the future, this will bring more opportunities for quality assurance experts to be utilized to their full potential.

Only time will tell what the future holds in store but based on current trends, expectations and conducted research we can summarize that the significance, valuation, quantities and automation of testing and quality assurance will increase in the coming years.

References

Agile Alliance. 2017. Available: https://www.agilealliance.org/glossary/atdd/. Accessed 18.3.2017.

Agile Manifesto. 2001. Available: http://agilemanifesto.org/iso/en/manifesto.html. Accessed 26.12.2016.

Berridge, C., Gray, A., Levy, R., Measey, P., Oliver, L., Roberts, B., Short, M., Wilmshurst, D. & Wolf, L. 2015. Agile Foundations. Principles, practices and frameworks. BCS, The Charted Institute for IT. United Kingdom. ISBN: 978-1-78017-254-5.

Black, R., Claesson, A., Coleman, G., Cornanguer, B., Forgacs, I., Linetzki, A., Linz, T., Aalst, L., Walsh, M., & Weber, S. 2014. Foundation Level Syllabus - Agile Tester. ISTQB. Available: http://www.istqb.org/downloads/send/5-agile-tester-extension-documents/41-agile-tester-extension-syllabus.html. Accessed 26.12.2016.

Buenen, M. & Muthukrishnan, G. 2016. World Quality Report 2016-17. Capgemini.

Bryman, A. & Bell, E. 2015. Business research methods. Fourth editions. Oxford University Press. United States of America. ISBN: 978-0-19-966864-9.

Cooke, J. 2016. Agile: An Executive Guide. Real results from IT budgets. Second Edition. IT Governance Publishing. United Kindom. ISBN: 978-1-84928-797-5.

Davis, J. & Daniels, K. 2016. Effective DevOps. O'Reilly Media, Inc. ISBN 978-1-4919-2630-7.

Edmondson, W. 2015. Agile is Dead. Available: http://www.williamedmondson.com/agile-dead/. Accessed 2.1.2017.

Eficode. DevOps. 2017.

Hanspal, S. 2014. Shift-Left Testing and overall Quality. Available: http://www.bitwiseglobal.com/blogs/shift-left-testing-and-overall-quality/ Accessed 11.6.2018.

Herschmann, J. & Murphy, T. 2016. Magic Quadrant for Software Test Automation. Gartner.

International Software Qualifications Board. ISTQB. Available http://astqb.org/glossary/. Accessed 18.3.2017.

Itkonen, J., Rautiainen, K. & Lassenius C. 2005. Towards Understanding Quality Assurance in Agile Software Development. Available: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=9B501F83D5FA2D4567AD14ABE9807DB5?doi=10.1.1.167.5055&rep=rep1&type=pdf. Accessed 18.12.2016

Kolose, A. 2015. DevOps adoption, challenges, & tools. Available: https://www.ca.com/en/blog-highlight/survey-devops-an-overnight-success-because-it-works.html. Accessed 23.9.2017.

Kyselynetti. 2017. Available: https://www.kyselynetti.com. Accessed 30.5.2017

Münch, J. & Mäkinen, S. 2013. Could-Based Software Engineering. Available: https://helda.helsinki.fi//bitstream/handle/10138/42910/cbse13_proceedings.pdf?sequence=2#page=49. Accessed 14.4.2017.

Saunders, M., Lewis, P. & Thornhill, A. 2009. Research Methods for Business Students. Fifth edition. Pearson Education Limited. England.

Shea, B. 2016. World Quality Report 2016-17. Capgemini.

Software testing fundamentals. 2017. Available: http://softwaretestingfundamentals.com/software-quality-assurance/. Accessed 18.3.2017

Solinea. DevOps 2017. Available: https://www.solinea.com/open-infrastructure-services/devops. Accessed 11.6.2017

Topham, B. 2013. DevOps and OpsDev: How maturity Model Works. Available: https://community.saas.hpe.com/t5/All-About-the-Apps/DevOps-and-OpsDev-How-Maturity-Model-Works/ba-p/306787#.Wby7SK0gn-Y. Accessed 16.9.2017.

Verona, J. 2016. Practical DevOps. First edition. Packt Publishing Ltd. UK. ISBN 978-1-78588-287-6

Waayenburg, H. 2016. World Quality Report 2016-17. Capgemini.

Walgude, A. 2016. World Quality Report 2016-17. Capgemini.

Vashishtha, S. 2017. Demystifying DevOps : Difference between Agile and DevOps. Available: http://www.agilebuddha.com/agile/x-htm/. Accessed 19.3.2017.

Wagner, A. 2016. Something has to change… Shift Left Shift Right or do BOTH!. Available: https://alwagnerblog.wordpress.com/2016/08/12/something-has-to-change-shift-left-shift-right-or-do-both/. Accessed 14.4.2017.

Wagner, A. 2015. What is Shift Left Testing? Available: https://www.ibm.com/developerworks/community/blogs/rqtm/entry/what_is_shift_left_testing?lang=en. Accessed 14.4.2017

Appendices

Appendix 1. Kyselytutkimuskysymykset (suomeksi)

Vastaajan taustatiedot

1. Nykyinen työnkuva *

Valitsen parhaiten kuvaava vaihtoehto

- ○ Testauksen ja laadunvarmistuksen asiantuntija (esim. testaaja, tekninen testaaja)
- ○ Testauksen ja laadunvarmistuksen asiantuntija, vastaan tietyn alueen testauksesta (esim. testauspäällikkö)
- ○ Vastaan yhden tai useamman alueen testauksesta ja laadunvarmistuksesta ja alueen kehittämisestä (esim. projektipäällikkö, hankejohtaja)
- ○ Muu, mikä? _____

2. Työskenteletkö tällä hetkellä testaukseen- ja laadunvarmistukseen erikoistuneessa yrityksessä? *

- ○ kyllä
- ○ ei

3. Kuinka monta vuotta olet työskennellyt testauksen- ja laadunvarmistuksen parissa? *

- ○ 0 - 5 vuotta
- ○ 6 - 10 vuotta
- ○ 11- 15 vuotta
- ○ 16 - 20 vuotta
- ○ 21 - 25 vuotta
- ○ 26 - 30 vuotta
- ○ yli 30 vuotta

Laadunvarmistuksen asiantuntijan ominaisuudet tulevaisuudessa

4. Seuraavassa on lueteltu laadunvarmistuksen asiantuntijan tyypillisiä ominaisuuksia. Kuinka tärkeänä näet seuraavat ominaisuudet tulevaisuudessa? *

	Vähäinen	Melko vähäinen	Tärkeä	Erittäin tärkeä	EOS
Liiketoiminnan ymmärtäminen	◯	◯	◯	◯	◯
Asiakkaan tarpeiden ymmärtäminen	◯	◯	◯	◯	◯
Tekninen osaaminen	◯	◯	◯	◯	◯
Testauksen automatisointi	◯	◯	◯	◯	◯
Virhekorjausten nopea edistäminen	◯	◯	◯	◯	◯
Testauksen edistymisen raportointi	◯	◯	◯	◯	◯
Viestintä- ja kommunikointitaidot	◯	◯	◯	◯	◯
Jatkuva osaamisen kehittäminen	◯	◯	◯	◯	◯

5. Mitä (muita) ominaisuuksia toivot laadunvarmistuksen asiantuntijalta tulevaisuudessa?

Ohjemistokehitys testauksen näkökulmasta

6. Kuinka tärkeänä näet seuraavat asiat tulevaisuudessa? *

	Vähäinen	Melko vähäinen	Tärkeä	Erittäin tärkeä	EOS
Toimiva vaatimustenhallinta	○	○	○	○	○
Testausympäristöt on virtualisoitu	○	○	○	○	○
Versionhallinta tukee testausta	○	○	○	○	○
Testaus on automatisoitu eri tasoilla	○	○	○	○	○
Automaattinen laadunvarmistus on yhdistetty vaatimuksiin	○	○	○	○	○
Uusi versio voidaan julkaista täysin automaattisesti	○	○	○	○	○
Palvelinympäristöt ovat pääosin virtualisoitu	○	○	○	○	○
Laadunvalvontaa voidaan seurata koko organisaatiossa	○	○	○	○	○

7. Mitkä ovat Sinun mielestäsi tärkeimmät asiat sovelluskehityksen laadunvarmistuksen kannalta tulevaisuudessa?

Testauksen automatisointi tulevaisuudessa

8. Kuinka tärkeänä näet seuraavien testaustyyppien automatisoinnin tulevaisuudessa? *

	Vähäinen	Melko tärkeä	Tärkeä	Erittäin tärkeä	EOS
Yksikkötestaus	○	○	○	○	○
Integrointitestaus	○	○	○	○	○
Järjestelmätestaus	○	○	○	○	○
Muuttumattomuustestaus	○	○	○	○	○
Hyväksymistestaus	○	○	○	○	○

9. Kuinka kattavasti automatisointi tulisi tehdä tulevaisuudessa? *

- ○ Ei lainkaan
- ○ Pienehkö määrä
- ○ Valtaosa
- ○ Kaikki mahdollinen/kannattava

10. Kuinka tärkeänä näet testausympäristöjen automatisoinnin tulevaisuudessa? *

	Vähäinen	Melko tärkeä	Tärkeä	Erittäin tärkeä	EOS
Testausympäristön automatisointi	○	○	○	○	○

Näkemyksesi tulevaisuuden toimitusfrekvenssistä

11. Mikä on mielestäsi sopiva julkaisuväli uusille ohjelmaversioille tulevaisuudessa? *

Valitse sopivin vaihtoehto

- ○ Päivittäin tai nopeammin
- ○ Viikoittain
- ○ Kuukausittain
- ○ Neljännevuosittain
- ○ Puolivuosittain
- ○ Vuosittain

12. Tuoko nopea julkaisuväli mielestäsi kilpailuetua tulevaisuudessa? *

- ○ kyllä
- ○ ei

13. Kokemuksesi mukaan kuinka hyvin laadunvarmistus vastaa mielestäsi tulevaisuuden tavoitetilaa? *

Ei vastaa tavoitetilaa — Vastaa täysin tavoitetilaa

14. Viitaten edelliseen kysymykseen mitkä ovat Sinun mielestä suurimmat esteet tavoitetilan toteutumisen suhteen?

15. Mikä on mielestäsi tärkein toimenpide, jolla laadunvarmistusta voidaan parantaa?